KURZWEIL VOICESYSTEM
USER'S MANUAL

KURZWEIL AI

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_09707

p/n 30-00435-002 Rev 02
January 26, 1986

KURZWEIL VOICESYSTEM VERSION 2.1

SOFTWARE RELEASE NOTES

Topics covered in release notes include

1. Version 2.1 KVS Software Release Kit Contents
2. Software Installation Instructions
3. Features of this release
4. Usage notes
5. Documentation for New Features
6. Known Problems
7. Bugs Fixed since Release 2.02

KIT CONTENTS

Your KVS Version 2.1 software release kit includes

- Cover letter
- Release Notes

  New KVS Users should read "Software Installation Instructions," below and then become familiar with the KVS User's Manual. The Release Notes describe changes in performance since Release 2.02 and undocumented aspects of KVS operation.

- Software: 4 IBM-PC/XT MS-DOS compatible floppy disks labelled as follows:

  1. KVS System Software, Disk 1 of 3

2. KVS System Software, Disk 2 of 3

3. KVS System Software, Disk 3 of 3

4. WORDKIT And Examples Disk

   This diskette contains some pre-typed language models useful as a starter kit for making your own language models. Two examples of language models which you can try and learn from are also on the disk. The examples are described in this document under "Documentation for New Features."

SOFTWARE INSTALLATION INSTRUCTIONS

Complete instructions for installing this software release are given in this document and in Chapter 2 of the KVS User's Manual. The install procedure differs slightly from Version 2.02 in that you must now reboot the KVS (as well as your PC) after the Install program completes, before you run KVLOAD.

Also, the Version 2.1 Install program detects problems with communications hardware and gives diagnostic messages when that happens (see "INSTALL PROCEDURE," below).

KVS Software Version 2.1 is completely compatible with DOS 3.x and can also run under DOS 2.x. An occasional problem with DOS 2.x is described in this document under "KNOWN PROBLEMS".

A NOTE ON UPWARD COMPATIBILITY FOR USERS OF VERSION 2.02

KVS Software Version 2.1 replaces Version 2.02. Any phrase books you made under Version 2.02 will be compatible with Version 2.1 software except for translations containing the <!wait> command. Those translations that use <!wait> to delay execution of pending KVS commands must now use <wait> - without the initial exclamation point (!) (see also "USAGE NOTES").

In addition, the KVS now requires a PC host system with a hard disk, or a 720K floppy. Since the KVS-resident software image is larger than 360K, you can no longer download the KVS from a dual-floppy PC host that has no hard disk.

If you have developed a voice application that uses the KVS Runtime Library, it currently still requires KVS software version 2.02. In that case, do not install the new release; contact your Kurzweil AI sales representative.

INSTALL PROCEDURE

Use the following procedure to install Release 2.1:

1. Boot your PC and get the MS-DOS C> prompt on the screen

2. Load the files from floppy disks. Insert the Disk 1 of 3 in drive A and then type

   A:INSTALL [RET]

3. Respond to the Install program's prompts. When prompted, remove disk 1 and insert disk 2. Press any key and the files on disk 2 will be installed. Repeat for disk 3 when prompted.

4. INSTALL then checks for potential problems with communications equipment. This is an interactive procedure. Since the Install program cannot tell how the hardware is connected, it reports what it can, and you must verify the hardware connections.

   Install tells you the port it will use for KVS-PC communications (the highest numbered port):

   "THE HIGHEST SERIAL PORT AVAILABLE FOR THE KVS IS COMn"

   where 'n' is 1 or 2. Be sure the RS-232 cable between the KVS and your PC is plugged into the COM port mentioned in the message. If it is go on to step 5 of these instructions. If you can't use that port, read on.

   (a) If that COM port on your PC is already in use (for network connections, an external modem, etc.) and you have another port available, you can set an environment variable called KVPORT to tell the KVS driver to read data from the low numbered port.

   When your PC has two ports and you connect the KVS to

COM1, you must add the command "SET KVPORT=1" to your autoexec.bat file. Install reminds you to give this information to KVS software:

"ADD SET KVPORT=1 TO YOUR AUTOEXEC.BAT FILE"

Use EDLIN or any other text editor to change AUTOEXEC.BAT.

(b) If your autoexec sets KVPORT, Install checks whether the specified port exists, and assigns the KVS to that port. If the port does not exist, you will see this message:

WARNING: KVPORT IS SET TO 'N' BUT THERE IS NO PORT AT THAT ADDRESS

Fix the "SET KVPORT=" command, or delete it from your autoexec file.

(c) Finally, Install tells you if your PC has no communications card. The message is:

"WARNING: THERE ARE NO COMMUNICATIONS PORTS CURRENTLY AVAILABLE"

To use the KVS, your equipment must have a serial port available to dedicate to KVS communications.

Install cannot detect the case where you have the wrong communications card plugged in (such as an ethernet card).

If message (a) or (c) shows up, KVLOAD will not work until you take remedial action.

5. After the equipment check, the screen will show the following message and some explanatory text:

"Deleting redundant Voice Profiles,"

If you are a new KVS user, type [Y] (yes) in response to this question.

If you are already a KVS user and haven't done so yet, you can save disk space and phrase book download time if you type [Y]. An answer of [Y] deletes the redundant copies of your Voice Profile from your phrase books.

Phrase books made before version 2.02 each contain a Voice Profile copy. More recent phrase books share a Voice Profile.

6. Reboot your computer. For an IBM PC XT or AT hold down [CTRL] and [ALT] and press [DEL].

7. Reboot the KVS. Find the power switch at the rear of the KVS and turn it off and on again.

8. At the c> prompt type

   KVLOAD [RET]

9. To install "WORDKIT AND EXAMPLES," insert the disk into drive A and type

   A: INSTALL [RET]

   See Chapter 2 of the KVS User's Manual for a description of WORDKIT.

These are the KVS directories created on your PC's fixed disk after INSTALL has run:

```
KVPROG <DIR>                    KVSLIB  <DIR>

KVSMOVE.BAT                     KV3
KV3DR.SYS                       KVS.KRL
KVCNTRL.EXE                     ENROLL.HLP
KVENROLL.EXE                    KVSV67.KFC
KVLOAD.EXE
KVMAKEPB.EXE                    KVSLMOD <DIR>
KVRESYNC.EXE
KVSEND.EXE                      ENROLL.KLM
KVTRAIN.EXE                     ENROLL.KMP
KVTEST.EXE
KVMAKELM.EXE
KVMODPB.EXE                     KVSUSERS <DIR>
DMA64.COM                       (no files)
KVLISTLM.EXE
```

Also, INSTALL will insert the line "device=C:kVPROGkV3DR.SYS" in your CONFIG.SYS file if it's not there already. If you have no path defined, INSTALL will append the line "path=C:kVPROG" to your autoexec.bat file.

FEATURES OF RELEASE 2.1

Improvements in this release are based on feedback from KVS customers, made in response to problems reported since Release 2.02. These fixes are the focus of this release.

Specific improvements have been made in the areas of KVS relilability, and error reporting. Several bugs have been fixed, and two new features added to keep recognition synchronized with the application task:

1. Application synchronization features:

   - The voice correction commands (#delete last word, #next choice, #try again) now restore the subset the KVS was in before you spoke erased phrase.

   - A new KVS command, <subset #previous>, lets a user perform application tasks in random rather than fixed order. When <subset #previous> appears in a translation string, the KVS returns control from the present subset to the one that was previously active. See also "Documentation for New Features," below.

2. Users can now print the contents of the PC screen by voice command.

3. Users can now press the shift keys by voice.

4. Developers can better synchronize voice control with slow running PC applications, such as IRMA.

5. To analyze poor recognition, developers can run the KVS in <output measurement>. This mode produces statistics on recognition that help determine whether deafness is due to thresholding, or poor candidate selection. Measurement mode is documented under "Documentation for New Features," below.

6. A new technique for automating user start-up with KVSEND is documented under "Documentation for New Features."

USAGE NOTES

This section describes

1. Testing recognition after enrollment
2. Enhancements in KVS recognition performance,
3. Improvements in language model making
4. Improvements in phrase book saving.

1. Testing recognition after enrollment

After enrolling, new KVS users should try recognizing the ENROLL vocabulary with the procedure described in Chapter 5 of the "KVS User's Manual." Testing recognition includes turning the microphone on and off, and correcting misrecognitions by voice command. Users can ask the KVS to respond differently to the same word, either typing the word in uppercase or lowercase, or typing a dictionary definition when the word spoken denotes a living thing. An excerpt of the language model that supports these features is shown below. The excerpt is provided for vocabulary developers new to the KVS to show the power of a KVS language model. Additional language model examples are given on the WORDKIT disk and in Chapter 8 of the KVS User's Manual.

LANGUAGE MODEL EXAMPLE:

A	B	C	D	E
PHRASE	UPPER	LOWER	LIVING	DEFINE_LIVING
drop	DROP\s	drop		
measure	MEASURE\s	measure		
song	SONG\s	song		
apple	APPLE\s	apple	apple	A round, firm, flesh...
termite	TERMITE\s	termite	termite	Any of the order (ISO..
flowers	FLOWERS\s	flowers	flowers	The seed-producing s..
new line	\r	\r	\r	\r
#TRY AGAIN	\b\s\b	\b\s\b	\b\s\b	\b\s\b
*SUFFIX		\s	\s	\s
*PREFIX				DEFINITION:\s

2. Performance enhancements

(a) Delayed output of recognized words (also known as the "one-behind" problem) no longer occurs.

(b) Deafness caused by near-capacity Speech Processing now causes the KVS to suspend recognition and resume when processing catches up. Messages are displayed in each case.

When using a phrase book that has as many tokens as the KVS can handle, you may see a message saying "Recognition temporarily suspended. Please stand by...". If you see this message, don't expect to be recognized until you see a second message saying "Recognition resuming." Formerly, no messages were reported.

Should you get the "recognition suspended" message in <edit> or another KVS angle bracket command, you can still type, but you can't make entries by voice.

To see how close your phrase book is to capacity, type <train {phrase book}>, select [A] add tokens, and press [P] to select the Pause menu. (See also "The KVS User's Manual," page 10-5.) To prevent frequent suspension of recognition, make your phrase book smaller by deleting extra tokens or phrases.

(c) Deafness caused by processing large phrase books in high background noise no longer occurs.

The message "Recognition temporarily suspended..." also appears when the KVS is working with very large phrase books and has heard its limit of continuous speech or noise for the current token resolution. Take the actions described in item (b) and check background noise level. Reduce the level of background noise by making sure people around you speak softly or move out of range of the microphone. See that the microphone is 1/4" away from your lips.

(d) Tokens too long to be recognized are now identified and refused during training.

If a phrase you say during training is too long for the KVS, you will see the message "TOKEN TOO LONG" and the token will not be stored. In that event, substitute a shorter phrase. For example, to the prompt

SAY: KAI

say "Kurzweil AI" instead of "Kurzweil Applied Intelligence" and remember to say the phrase the same way in recognition as when adding tokens. (See also "The KVS User's Manual," page 9-5.)

The maximum token length at each value of token resolution is given below:

TABLE 1 -- Maximum token length per value of token resolution

Utterance duration (seconds)	Token resolution	
4.5	1	
4.0	2	
3.5	3	
3.0	4	(the default)
2.5	5	
2.0	6	
1.5	7	

(e) The KVS correctly handles phrases with tied recognition scores. One phrase is typed; the other will be typed when the user says "#next choice."

(f) The KVS no longer delays before recognizing the first word after a microphone check.

(g) If the KVS is not on line or downloaded when you type KVTEST, you are prompted to type KVRESYNC or KVLOAD.

3. Language Model Making

(a) <edit> protects against accidental invocation of Transparent Mode [CTRL-T]. Transparent Mode, also known as Learn Mode, teaches <edit> the translation of a phrase.

Now whenever <edit> receives a [CTRL-T] keypress, the window says:

> Press [RET] to use Transparent Mode or [ESC] to escape.
> Press [CTRL-T] to end use of Transparent Mode.

Press [RET] to confirm your choice of Transparent Mode, or press [ESC] if you have entered Transparent Mode accidentally. [ESC] returns to the <edit> menu.

If you confirm your choice of Transparent Mode, the <edit> window disappears as you type to your application. <edit> saves the keypresses you type and stores them in the translation string you are editting. When you finish the translation, type [CTRL-T] again to end use of transparent mode and return to the <edit> menu.

(b) KVLISTLM accepts (but ignores) an extension (.klm) on the input filename.

(c) Language model overflow is handled better.

<Edit> gives an error report when any of the following conditions is true:

1. The language model is greater than 128K.

2. New translation strings, phrase names and subset names together exceed available working storage.

3. The language model has more than 50 subsets.

4. There is no more room in working storage for translation strings.

The message is:

THERE IS NO MORE ROOM FOR PHRASES, SUBSET NAMES AND TRANSLATION STRINGS.TRY SAVING TO DISK OR DELETING SOME PHRASES AND SUBSETS PRESS [ret] TO EXIT TO MAIN MENU

If you see this message, try writing the language model or phrase book to disk to relieve the overflow condition. <Edit> condenses the language model space before writing it to disk and this may help. Deleting phrases and subsets, if possible, will certainly help.

(d) <Edit> accepts the key sequence CTRL-F3 in translation strings.

(e) <Edit> accepts eight bit characters in translation strings and phrase names.

(f) The command <window> in a translation string erases a KVS window from the screen.

(g) To display a multi-line message in the KVS window, use \n as a line terminator. For example, the translation <window Line1\nLine2\nLine3> displays a three line window.

(h) The function of <!wait> command in a translation string has been modified.

Formerly, the <!wait> command included an exclamation point (!) to delay output of typeahead and execution of waiting angle bracket commands. Now, if you want to delay execution of waiting angle bracket commands, you must change the form of the command to:

<wait>

without the !. For example, to display a subset name on the screen for one second, you'd use this translation string:

<subset> <wait 1000> <window>

<subset> puts the subset name in the KVS window, <wait 1000> waits one second before sending <window> to the host PC. <Window> then erases the window.

This change is only for delaying execution of normal angle bracket commands. You must still use <!wait> to delay sending of typeahead or instant angle bracket commands -- those containing the exclamation point in the command such as <!status>.

(i) Translation strings can now have up to 12 angle bracket commands instead of 6. This means that the error message "Too many commands waiting" appears less often.

(j) To print the contents of the PC screen by voice command, use the translation \0\162 in your language model. For instance:

PHRASE	TRANSLATION
Print screen	\0\162

You can enter this translation in <edit> just by pressing [ctrl] and [prt scr].

(k) To synchronize spoken commands with slow running PC applications issue the following command at the C> prompt before starting the application:

KVCNTRL DELAY nnn

where 'nnn' is a decimal number indicating milliseconds. This command delays typeahead output to programs by 'nnn' milliseconds per character. This command is particularly useful for some IRMA users where unsolicited input is thrown out. You must determine the duration of the delay by trial and error. KVS software rounds up the number you specify to the nearest 55 milliseconds, corresponding with the granularity of the PC's clock.

(1) To allow users to press shifted keys by voice, send the KVS the same character string two separate times. The first sequence presses the shift key, the second releases it. For instance, to invoke Sidekick from the keyboard you press [CTRL] + [ALT]. To invoke Sidekick by voice, use the following translation string:

PHRASE	TRANSLATION
sidekick	0\376\14\1 <!wait 300> 0\376\14\1

The <!wait> command is included because Sidekick wants the key depressed for a certain time so it can detect it.

In the translation string, the first two bytes are used to specify a shifted key, and and the second two bytes the particular keys being depressed. Use this table to code your translation:

Byte 1

CODE	TRANSLATION
10	[ALT]
4	[CTRL]
2	[left shift]
1	[right shift]

Byte 2

CODE	TRANSLATION
200	[INS]
100	[caps lock]
40	[num lock]
20	[scroll lock]
10	[suspend/ AT keyboard only]

1 none of the above

If the byte represents a sequence of keys, instead of a single key add the values of the keys to be depressed. See Sidekick example above, where [CTRL]=4 and [ALT]=10 are added to form the value of Byte 1, which is 14.

4. Phrase Book Loading and Saving

Any time a phrase book is saved, all modified parts of it are saved (except tokens when publishing).

(a) The KVS checks available disk space before a phrase book is saved.

If you'll run out of disk space in the middle of a save, the KVS won't start the save operation. Instead you'll see the message:

```
PHRASE BOOK SAVE IS ABORTED DUE TO INSUFFICIENT DISK SPACE.
PLEASE FREE UP SOME DISK SPACE AND TRY THIS COMMAND AGAIN.
PRESS [ret] TO CONTINUE
```

You can get this message (a) when you choose [S]ave from a <train> or <edit> menu, (b) at sign off when you type <bye> and save a changed phrase book, or (c) when starting recognition with a new phrase book while an old phrase book is loaded. A second message window will tell you the context of the save, for instance:

```
FAILURE PUBLISHING PHRASE BOOK/ PRESS [ret] TO CONTINUE
FAILURE SAVING PHRASE BOOK/ PRESS [ret] TO CONTINUE
```

(b) The KVS saves large phrase book correctly.

When a phrase book is too big to CONDENSE, you will see the following message:

```
FAILURE - LANGUAGE MODEL IS TOO BIG TO CONDENSE
NON-CONDENSED LANGUAGE MODEL IS RESTORED
PRESS [RET] TO CONTINUE
```

Should you get this message, the pre-condensed version of the phrase book is saved. Delete some phrases, tokens or subsets and try the save again.

Though most phrase books are saved in two passes (condensing and saving), a large phrase book with an application-specific voice profile can take up to four passes to save. That is, dots will paint across the screen up to four times.

(c) The keyboard does not lock up during phrase book saves.

(d) Tokens sets are checked for completeness and consistency when a phrase book loads.

When a phrase book loads, the message "CHECKING TOKEN SET" appears and the load operation takes slightly longer. A phrase book that fails the data check produces this message:

```
TOKEN SET FOR PHRASE BOOK {name} HAS BEEN CORRPUTED
CORRUPTED TOKEN SET WILL BE SAVED AS "BOGUS.KVX"
PRESS [RET] TO CONTINUE
```

You must press [ret] at this point. The corrupted phrase book is cleared from memory. Any changes since the last save are lost. If you get this message, call Kurzweil Customer support.

DOCUMENTATION FOR NEW FEATURES

Features new with this release include

1. The <subset #previous> command
2. The Example language models on the Wordkit disk,
3. <output measurement mode>
4. The use of KVSEND with a batch file

This section provides preliminary documentation for these features, since they are not documented in the KVS User's Manual.

1. <subset #previous>

   Use of this command in a translation string lets you perform application tasks in random, rather than fixed order. The KVS remembers the subset that was active before a subset change, and returns there if it gets a <SUBSET #PREVIOUS> Command.

   The language model example below lets radiologists dictate reports while examining x-ray films. To make a report, the radiologist can dictate results of a mammogram (subset mammog), an ultrasound exam (subset ultra), or simply say sentences about the patient (subset FTEXT). The <subset #previous> command lets the doctor move back and forth among these three subsets just by saying "change exam." This allows the radiologist to dictate the report freely and avoids a fixed order.

Language Model Example:

A	B	C	D
PHRASE	MAMMOG	ULTRA	FTEXT
FREE TEXT MODE	<subset ftext>	<subset ftext>	
CHANGE EXAM	<subset #previous>	<subset #previous>	<subset #previous>
NORMAL	\013r\rmam.n	\013r\rultra.n	normal
ROUTINE	\013r\rmam.r	\013r\rultra.r	routine

The KVS error correction commands -- #delete last word, #next choice, #try again -- now automatically excecute a <subset #previous> restoring the application to the state it

was in before the erased word was spoken.

2. The Example Language Models

The Diskette labeled "Wordkit and Examples" includes two example language models which you can try out and learn from.

TRADING shows an example of trading commodities by voice. It demonstrates the use of multiple subsets, help windows, correction by voice, and automatic subset switching.

STARTUPH provides vocabulary for executing most of the KVS commands by voice. It can be used as a base for building a language model for hands-free, remote, and handicapped applications.

INSTALLATION: The examples are located on the diskette labeled "WORDKIT and EXAMPLES", but are NOT installed by the Wordkit installation procedure. You must copy them from the floppy into the \KVSLMOD directory yourself. At the C:> prompt, type

```
copy a:\kvslmod\trading.*  c:\kvslmod
copy a:\kvslmod\startuph.* c:\kvslmod
```

LISTING: After copying the language models, you can print them by typing at the C> prompt

```
cd \kvslmod
kvlistlm -a startuph >prn:
kvlistlm -a trading >prn:
```

USING THE TRADING LANGUAGE MODEL

The TRADING Language Model is similar to the language model diagrammed on page 13-3 of your User's Manual. It is composed of four subsets: ACTION, QUANTITY, COMMODITY, and EXCHANGE and automatically switches between the subsets as you speak sentences like "Buy five wheat kansas". Automatic subset switching is implemented with the *SUFFIX command described on page 8-17 of your manual.

Once the demo is trained you say the words listed in Table 2.

TABLE 2. Phrases you can speak in the TRADING Demo

SAY	KVS DOES:
In ANY subset:	
"trading demo"	starts kvtest, prints instructions
"end of demo"	exits kvtest
"new transaction"	go to subset ACTION
"What can I say"	shows current subset
"next subset"	goes to next subset
"previous subset"	goes to previous subset
"microphone off"	turns mic off, displays window.
"listen to me"	turns mic on, displays window.
"next choice"	delete word, type next best guess.
"try again"	delete word, don't type it again
"delete last word"	delete word.
In subset ACTION:	
"Buy"	types "Buy ", goes to QUANTITY
"Sell"	types "Sell ", goes to QUANTITY
In subset QUANTITY:	
"1",..."12"	types "1 ", goes to COMMODITY
In subset COMMODITY:	
"corn", "wheat", etc.	types "corn", etc. goes to EXCHANGE
In subset EXCHANGE:	
"Chicago", "New York"	types city and goes to subset ACTION

USING THE STARTUPH LANGUAGE MODEL

For hands-free start-up of the KVS using STARTUPH, copy the STARTUPH language model into \KVSLMOD and train it. Then design an autoexec.bat file like the following example:

AUTOEXEC.BAT File for Use with STARTUPH

```
path C:\;C:\DOS;C:\KVPROG
KVLOAD/N
KVSEND {USER PAUL}
KVSEND {REC STARTUPH}
```

This language model has been designed to approach hands-free operation of the KVS. It is composed of six subsets: ONCE, COMMAND, #VOICE_COMMAND, #EDIT, ALPHA and DIGITS.

Automatic subset switching is implemented with *ENTER in subset ONCE. The translation for *ENTER performs an automatic microphone check and switches to the COMMAND subset.

The COMMAND subset contains some useful DOS commands and phrases to run the KVS host utilities. It also allows you to say "angle-bracket" to pull up the KVS window.

The #VOICE_COMMAND subset is automatically active when the KVS window is on the screen. It allows you to say all the KVS angle bracket commands.

#EDIT lets you create and edit language models by voice. Study the listing KVLISTLM produces to learn the phrases you can use.

The ALPHA and DIGITS subsets can be used when you are building new language models with <EDIT>.

All subsets allow you to do a microphone check and adjust microphone gain by voice.

All subsets also allow you to add a word by spelling it with the phonetic alphabet. Here is an example of adding the word "HI":

Example: To add "HI" to the vocabulary, say:

```
"add a word"
"hotel"
"india"
"train new word"
"hi"
"hi"
```

All subsets allow you to add or replace tokens of a word. Here is an example of adding and replacing tokens of "HI":

Example: To add or replace tokens of "HI", say:

```
"train tokens"
 "add token"    (or " replace token")
 "hotel"
 "india"
 "training"
 "hi"
  .
  .
```

A few phrases in STARTUPH will lead to manual keypresses. These are "add phrases" and "restart". All other phrases will operate completely by voice command.

3. <output measurement>

Measurement is a new KVS <output> mode that lets you analyze problems with recognition. To use measurement

- Exit your application and run KVTEST. Type at C>

  KVTEST [ret]

- Be sure the KVS is loaded with the phrase book you want to evaluate; then put the KVS in measurement output mode:

  <recog {pbname}> **optional
  <output m t>

- Say the phrase that has been misrecognized or not heard.

- View the results. The screen will show a list of tokens that match your phrase. The list is ordered by token index number, and for each item, the raw score and distance measure are printed. A low distance and a high score indicate a good match.

  Here is partial output for the phrase "bat". Five of the 30 candidates found by the acoustic matching process are shown:

Sample output: Measurement Mode

Duration 0.48
30 Candidates

Generation 720.64

Index	Dist	Score	Spelling
6	58	436	hat
52	55	562	ent
86	54	578	back
89	52	585	bank
214	16	846	bat
.			
.			
.			

- Evaluate the output:

  (1) Check the duration of the utterance. The token length in seconds is printed in the summary information preceding the statistics (here .48 seconds). A reasonable length ranges from .35 to .85 seconds. The average phrase is .6 seconds. A phrase that takes longer than a second to say may be cut short and misrecognized.

  (2) Be sure the phrase you said has a high score and a low distance. A phrase recognized with high confidence will have a score approaching or over 1000, and a low distance. "Bat," for example, has a score of 846 and a distance measure of 16.

  (3)See whether any other phrases score closely. Pairs of phrases that are confusable will have close, or identical

scores. In that case, you can decrease the confusion by changing one of the pair to a synonym, when possible, or use subsets.

(4)If the KVS was deaf, see whether all the listed tokens scored below rejection threshold. In that case, you can lower the rejection threshold See <thresh> and <parameter bscore> in the KVS User's Manual.

Although measurement mode shows all tokens found by acoustic matching, normal recognition does not use all of the listed candidates. Tokens eliminated in normal recognition include phrases not legal in the active subset and tokens scoring below the recognition threshold. Such tokens never become recognition candidates.

- To end measurement mode, return the KVS to normal recognition by typing

<output t t>

- Exit KVTEST by typing [CTRL-Z].

4. KVSEND can now take input redirected from a file. Given a file called startup.txt containing the following KVS commands:

```
<user john>
<threshold 3>
<recognize demovr>
```

You can now send those commands to the KVS by typing at C>,

```
KVSEND < startup.txt
```

This executes faster than the alternative technique, which is to write KVSEND commands in an autoexec.bat file: For example

in autoexec.bat

```
KVSEND {user john} {threshold 3} {recognize demovr}
```

Remember to quote the angle bracket (`<) in order to avoid opening the KVS window.

KNOWN PROBLEMS

When the KVS sends voice input to certain host-resident programs running under DOS 2.x, contention for operating system services may cause the PC to beep. If it does, you must reboot both the PC and the KVS. This problem can occur with Sidekick (tm Borland International). Avoid running Sidekick with DOS 2.x. Use DOS 3.x instead.

BUG FIXES

1. A Bug in DOS 2.0 and 2.1 for the IBM XT

   A program that fixes a bug in DOS 2.0 and 2.1 is supplied in \KVPROG with this release. The program is called DMA64 and is installed automatically only if you're using Version 2.0 or 2.1 of DOS. Appendix C of the KVS User's Manual describes the problem and the fix DMA64 provides.

2. Off by 1

   A bug in Release 2.02 sometimes caused the next phrase in the phrase book to be typed instead of the high scoring phrase. This bug is fixed in the new releasse.

3. #next choice string handles "cancel affix" correctly.

   In Release 2.02, translation strings containing \\a (cancel affix) did not work properly when the user said "#next choice." The KVS would sometimes send the string \\a as part of the translation. This has been fixed in the new release.

4. KVLISTLM now takes filenames with a path specification for the current directory. That is you can now type

   ```
   KVLISTLM .\filename
   ```

5. KVLISTLM now processes the -c switch (comma separated list) without crashing.

6. The KVS no longer gives an "inconsistent token set" message when the PC is loading or saving a phrase book and doing application I/O at the same time.

7. KVMODPB no longer has a stack overflow problem. In version 2.093, this problem required you to type KVMODPB=30000 to modify large phrase books. You no longer have to type "=30000".

8. Saving tokens during enrollment now works. Whenever enrollment is paused, you can explicitly ask for tokens to be saved by choosing [S] from the enroll menu.

9. KVMAKELM now rejects subset names containing spaces when it finds them in a .prn file.

10. KVMAKELM no longer fails when compiling import format language models.

RELATED PUBLICATIONS

"The KVS Reference Manual" provides detailed documentation on the KVS Run-Time Library, the KVS <angle bracket> commands, and host utilities. The KVS Run-Time Library is designed for application programmers who wish to integrate voice input using the KVS with new or existing applications programs.

"The KVS Porting Manual" contains documentation to allow software engineers with experience in writing device drivers to port the KVS host software to other operating systems and/or host hardware. The manual describes the design, testing and operation of the KVS host-resident software by using Kurzweil's current implementation on the IBM PC/XT/AT running PC-DOS as an example.

"The KVS Ready Reference Card" contains summaries of <edit> command functions, KVS <angle bracket> commands, and KVS host utilities.

The IBM Personal Computer "Disk Operating System", version 3.00 explains the syntax, format and function of DOS commands.

The IBM Personal Computer AT/XT "Guide To Operations" explains how to operate IBM AT/XT hardware.

The IBM Personal Computer AT/XT "Installation and Setup" guide explains how to install and test IBM AT/XT hardware.

For systems programmers, the IBM Personal Computer "Disk Operating System Technical Reference" Manual, Version 3.00, has information about screen and keyboard functions, device drivers, disk space allocation, and memory management. It also shows how an application program can send commands to DOS.

NOTES:

SUMMARY TABLE OF CONTENTS

ANALYTIC TABLE OF CONTENTS *iv*
LIST OF FIGURES *x*
LIST OF TABLES *xi*
LIST OF TASKS *xii*
FOREWORD *xv*

CHAPTER 1 - INTRODUCING THE KURZWEIL VOICESYSTEM

CHAPTER 2 - INSTALLATION AND STARTUP

CHAPTER 3 - SPEAKING TO YOUR COMPUTER

CHAPTER 4 - ENROLL: MAKING THE FIRST PHRASEBOOK

CHAPTER 5 - TRY RECOGNITION

CHAPTER 6 - THE PHRASEBOOK CREATION CYCLE: MAKE, TRAIN, TEST, CHANGE

CHAPTER 7 - TOOLS FOR MAKING LANGUAGE MODELS

CHAPTER 8 - LANGUAGE MODEL DESIGN

CHAPTER 9 - MAKE A PHRASEBOOK

CHAPTER 10 - TRAIN A PHRASEBOOK

CHAPTER 11 - TEST A PHRASEBOOK: MORE ON RECOGNITION

CHAPTER 12 - CHANGE A PHRASEBOOK

CHAPTER 13 - THROUGHPUT: TUNING THE KVS FOR BEST PERFORMANCE

CHAPTER 14 - TAILORING STARTUP AND DISTRIBUTING LANGUAGE MODELS

APPENDIX A - KEYBOARD CODES FOR THE PC & COMPATIBLES
APPENDIX B - COMMAND SUMMARIES
APPENDIX C - HARDWARE AND SOFTWARE COMPATIBILITY
APPENDIX D - KURZWEIL VOICE TERMINAL INSTRUCTIONS

INDEX

ANALYTIC TABLE OF CONTENTS

LIST OF FIGURES AND TABLES		*x*
LIST OF TASKS		*xii*
FOREWORD		*xv*
CHAPTER 1	INTRODUCING THE KURZWEIL VOICESYSTEM	1-1
	KVS CAPABILITIES	1-2
	HOST RESOURCES REQUIRED	1-3
	VOCABULARY CAPACITY	1-3
	INSIDE THE KVS	1-4
	SETTING UP FOR SPEECH: A SUMMARY	1-4
	THE SPEECH CONNECTION	1-6
	OVERVIEW OF THE KVS INSTALLATION PROCEDURE	1-6
CHAPTER 2	INSTALLATION AND START-UP	2-1
	UNPACKING INSTRUCTIONS	2-2
	SETTING UP THE KVS	2-2
	Access	2-2
	Power Requirements	2-2
	Temperature and Humidity	2-3
	Ambient Noise, Echoes and Small Rooms	2-3
	HARDWARE INSTALLATION INSTRUCTIONS AND CHECKLIST	2-3
	Making Cable Connections	2-4
	WILL THE KVS WORK WITH YOUR PC?	2-7
	INSTALLING SYSTEM SOFTWARE	2-7
	What INSTALL Does	2-7
	Installing WORDKIT	2-9
	If INSTALL Didn't Work	2-11
	KVS START-UP: POWER ON AND KVLOAD	2-11
	If KVLOAD Didn't Work	2-12
	KVSPOWER OFF	2-15
	KVS RESTART	2-15
CHAPTER 3	SPEAKING TO YOUR COMPUTER	3-1
	HEADSET ADJUSTMENT	3-2
	MICROPHONE POSITION	3-3
	Checking Microphone Position	3-3
	Caring for theHeadset Microphone	3-5
	SPEAKING TIPS	3-5
	What's a Phrase?	3-6
	Speaking Rate	3-6
	Pronunciation	3-7
	Clarity	3-7
	Volume	3-7
	Consistency	3-8
	Changes In Your Voice	3-8

CHAPTER 4 ENROLL: MAKING THE FIRST PHRASE BOOK 4-1

GETTING STARTED 4-2
YOUR VOICE PROFILE 4-2
Preview of KVENROLL 4-2
Recording Tokens 4-3
KVS Computation 4-3
MAKING TOKENS 4-3
STARTING ENROLLMENT: KVENROLL 4-5
"TESTING ONE TWO. . .": THE MICROPHONE CHECK 4-5
RECORDING TOKENS 4-7
INTERRUPTING RECORDING 4-9
Pausing Briefly 4-9
Replacing a Questionable Token 4-9
Saving Your Tokens 4-9
ENROLLING IN MORE THAN ONE SITTING 4-10
Resuming an Interrupted Enrollment 4-10
FINISHING ENROLLMENT 4-11
Checking Computation Status 4-11
IF ENROLL DIDN'T WORK 4-12
PHRASE BOOK FILES 4-13
PHRASE BOOK BACK UP: KVSMOVE 4-15
If KVSMOVE Didn't Work 4-15
Using DOS COPY 4-16

CHAPTER 5 TRY RECOGNITION 5-1

TESTING RECOGNITION 5-2
Start Up an Application Program 5-2
Sign On as a KVS User: <user> 5-3
Start Recognition: <recognize> 5-3
HOW TO USE ONE PHRASE TO TYPE MANY TRANSLATIONS 5-4
VOICE CORRECTION 5-6
STOPPING AND RESUMING RECOGNITION 5-6
Pause and Sign Off: <pause>, <bye> 5-7
Restarting Recognition after <pause> 5-7
Repeating or Bypassing the Microphone Check 5-8
MAKING A PHRASE BOOK FOR YOUR APPLICATION 5-9

CHAPTER 6 THE PHRASE BOOK CREATION CYCLE:

MAKE, TRAIN, TEST, CHANGE 6-1
WHAT IS A PHRASE BOOK? 6-2
WHAT IS A LANGUAGE MODEL? 6-2
THREE WAYS TO MAKE A LANGUAGE MODEL 6-3
The <edit> Command 6-3
A Spreadsheet Program 6-5
A Text Editor 6-7

CHAPTER 7 TOOLS FOR MAKING LANGUAGE MODELS 7-1

A SAMPLE LANGUAGE MODEL: NEWDATES 7-2
TUTORIAL: HOW TO USE <edit> TO MAKE A LANGUAGE MODEL 7-6
Step 1: Write Out the Phrases and Translations in the Vocabulary 7-7
Step 2: Sign On To the KVS and Invoke the <edit> Command 7-7
Step 3: Merge WORDKIT's Vocabulary Into Your Phrase Book 7-8
Step 4: Copy Subsets 7-9
Step 5: Edit Translations 7-11
Step 6: Add Phrases and Translations To The Language Model 7-11
Short Cuts 7-13
Wild Cards and String Completions 7-14
Step 7: Save the Language Model (Optional) 7-14
Step 8: Train the Language Model 7-15
USING WORDKIT WITH A SPREADSHEET PROGRAM 7-15
TUTORIAL: HOW TO USE A SPREADSHEET PROGRAM TO MAKE A LANGUAGE MODEL 7-16
Step 1: Change the KVS Command Delimiter 7-17
Step 2: Load WORDKIT into the Spreadsheet Program 7-17
Steps 3, 4: Add the Years of the Decade and the Names of the Months 7-17
Step 5: Verify Days of the Month (Number Words) in the PHRASE Column 7-17
Step 6: Add Three DOS Commands to the PHRASE Column 7-18
Step 7: Make Column E a Recognition Subset Labeled Dates 7-18
Step 8: Enter Translations (Arabic Numbers) for the Days of the Month at Cells E21..E31 7-18
Step 9: Add Translations for the Years at E35 7-19
Step 10: Add Translations for Months at E46 7-19
Step 11: Add Translations of Voice Commands and DOS Commands 7-19
Step 12: Save the New Language Model as a Print File, and Save the Worksheet 7-20
Step 13: Restore the KVS Command Delimiter 7-20
TUTORIAL: HOW TO USE A TEXT EDITOR TO MAKE A LANGUAGE MODEL 7-20

CHAPTER 8 LANGUAGE MODEL DESIGN 8-1

WHAT GOES INTO A LANGUAGE MODEL? 8-2
Choosing Spoken Phrases 8-2
Valid Phrases 8-3
Defining Translation Strings 8-4
Valid Translation Strings 8-4
Star and Sharp Phrases 8-6
Subsets 8-7
Multiple Meanings 8-8
Improved Accuracy 8-8
User Control Over Application Flow 8-8
Sharp Subsets: #edit and #voice command 8-8
HOW BIG CAN A LANGUAGE MODEL BE? 8-9
CONSIDERATIONS FOR LANGUAGE MODEL DESIGN 8-10
Requirements of the Application 8-10
Ease of Use 8-10
KVS Requirements 8-11
LANGUAGE MODEL EXAMPLES 8-12

CHAPTER 9 MAKING A PHRASE BOOK

THE FILES IN A PHRASE BOOK: LANGUAGE MODEL, VOICE PROFILE, TOKENS 9-2
HOW TO MAKE A PHRASE BOOK STARTING WITH <edit> 9-2
KVMAKELM: HOW TO MAKE A PHRASE BOOK STARTING WITH A SPREADSHEET PROGRAM 9-5
KVLISTLM: HOW TO PRINT A LANGUAGE MODEL 9-6
KVLISTLM: HOW TO RETURN A LANGUAGE MODEL TO ASCII FORMAT FOR EDITING WITH A SPREADSHEET 9-8
KVMAKEPB: HOW TO PUT AN UNTRAINED PHRASE BOOK IN THE END-USER'S DIRECTORY 9-9
KVMODPB: HOW TO USE EXISTING TOKENS WITH A NEW VERSION OF A LANGUAGE MODEL 9-10

CHAPTER 10 TRAIN A PHRASE BOOK 10-1

STARTING TRAINING WITH AN <edit> LANGUAGE MODEL 10-2
STARTING TRAINING WITH AN ASCII LANGUAGE MODEL 10-2
RECORDING TOKENS 10-3
SKIPPING TOKENS DURING RECORDING 10-4
PAUSING AND RESUMING RECORDING 10-5
Obtaining a Status Report on Training 10-5
SAVING TRAINING 10-5
ADDING, REPLACING, AND DELETING TOKENS 10-6
Adding Tokens 10-7
Replacing Tokens 10-8
Deleting Tokens 10-8
Count Tokens 10-8
If <train> Didn't Work 10-9

CHAPTER 11 TEST A PHRASE BOOK: MORE ON RECOGNITION

WHAT PERFORMANCE SHOULD YOU EXPECT OF THE KVS? 11-2
IF THE UNEXPECTED HAPPENS 11-3
FIND PROBLEM PHRASES: ORDERED OUTPUT MODE 11-4
Using Ordered Output Mode 11-5
CHECK SUBSET 11-7
ADJUST REJECTION THRESHOLD 11-8
DELAY KVS MESSAGE TRANSMISSION <!wait> 11-9

CHAPTER 12 CHANGE A PHRASE BOOK

WHY CHANGE YOUR PHRASE BOOK? 12-2
CHANGING THE WRITTEN PART OF YOUR PHRASE BOOK 12-2
<edit> FEATURES 12-5
Using the <edit> Command 12-6
Making a New Phrase Book from a Published Language Model 12-6
Making a New Phrase Book 12-6
Selecting an Item from the Menu 12-7
Help: List Editing Options 12-9
A Word About Wild Cards 12-12
Leaving <edit> 12-12
<edit> FUNCTIONS 12-13
Phrase, Subset and Translation Editing 12-13
Phrase Book File Mangement 12-21
Reporting Language Model Size and Contents 12-26
Set-up of Phrase Books for Data Collection 12-30
WHAT YOU HAVE LEARNED 12-33

CHAPTER 13 THROUGHPUT: TUNING THE KVS FOR BEST PERFORMANCE

ERROR PREVENTION: DESIGNING ERRORS OUT 13-2
Make Phrases More Distinctive 13-2
Remove Confusion by Subsetting 13-3
ERROR CORRECTION: IMPROVING TOKENS 13-4
For Highest Accuracy: Changing Token Resolution 13-4
IMPROVING ENROLLMENT DATA 13-5
Improving the Enroll Phonetic Model 13-5
Computing a Phonetic Model on the Application Vocabulary 13-6
Enrolling on an Application Vocabulary Instead of on the Enroll Vocabulary 13-7
HOW TO MINIMIZE EFFECTS OF BACKGROUND NOISE 13-8
In Training 13-8
In Recognition 13-9
TALKING FASTER: SPEEDING UP DIGIT ENTRY 13-9

CHAPTER 13 THROUGHPUT: TUNING THE KVS (continued)
VOICE MACROS: EXECUTING SEVERAL COMMANDS WITH A SINGLE PHRASE 13-10
VOICE CORRECTION 13-11
ADJUST PARAMETERS CONTROLLING THE CANDIDATE LIST 13-11
OUTPUT MODES: CUSTOMIZING OUTPUT 13-13

CHAPTER 14 TAILORING START-UP AND DISTRIBUTING LANGUAGE MODELS

WHAT IS AN AUTOEXEC.BAT FILE? 14-2
AUTOMATIC KVS INITIALIZATION 14-2
AUTOMATIC KVS START-UP 14-2
AUTOMATIC USER START-UP 14-3
Automatic User Start-up by Typing User Name 14-4
HOW TO CREATE A BATCH FILE 14-5
HOW TO STORE A LANGUAGE MODEL AND TRANSFER IT BETWEEN USERS 14-5
Storing a KVS Format Language Model on the Fixed Disk 14-5
Transferring an ASCII Language Model Via the Fixed Disk 14-6
Transferring a Language Model Via Floppy Disk 14-7

APPENDIX A IBM PC KEY CODES A-1

APPENDIX B COMMAND SUMMARIES B-1

Commands to Invoke Host Utilities B-1
Angle Bracket Commands for KVS Control B-3
<edit> Command Summary B-9
EDLIN commands B-11

APPENDIX C HARDWARE AND SOFTWARE COMPATABILITY

COMMUNICATION BETWEEN THE KVS AND A MAINFRAME C-1
CHECKLIST OF CRITICAL ITEMS FOR COMPATIBILITY C-1
POTENTIAL KVS CONFLICTS C-2
Communications Conflicts C-2
Keyboard Conflicts C-4
Irma C-4
Forte C-4
BUG IN DOS 2.10-2. FOR THE IBM XT C-4

APPENDIX D KURZWEIL VOICE TERMINAL (KVT): Special Instructions D-1

INDEX I-1

LIST OF FIGURES

Figure 1-1.	KVS Memory Use for Host RAM and Disk Space	1-4
Figure 1-2.	Desktop KVS, Headset, and IBM-PC AT/XT	1-5
Figure 2-1.	KVS Front and Back Panels	2-4
Figure 2-2.	Connecting KVS to Computer with 25-pinSerial Port	2-6
Figure 2-3.	Connecting KVS to Computer with 9-pin Serial Port	2-7
Figure 2-4.	The KVS Pop Up Window	2-12
Figure 2-5.	Mic/Line Switch on KVS Bottom Panel	2-15
Figure 3-1.	Microphone Placement	3-4
Figure 3-2.	The Adjustable Headset	3-4
Figure 6-1.	A Phrasebook With its Three Parts	6-3
Figure 6-2.	Phrasebook Creation Using <edit>	6-4
Figure 6-3.	Phrasebook Creation Using External Editor Program	6-5
Figure 7-1.	Export.prn File Format of Dates Language Model	7-21
Figure 9-1.	Sample Language Model Printout	9-7
Figure 9-2.	Import.prn File Format for 1-2-3 Import Files	9-9
Figure 12-1.	Phrase Book Modification Cycle Using <edit>	12-3
Figure 12-2.	Phrase Book Modification Cycle Using Spreadsheet	12-4

LIST OF TABLES

Table 2.1.	Contents of the Wordkit Demo Disk	2-9
Table 4.1.	The Balanced Vocabulary	4-4
Table 4.2.	Enroll Phrase Book Files	4-14
Table 5.1.	Living Things in the ENROLL Vocabulary	5-5
Table 5.2.	Turning Off and Resuming Recognition	5-7
Table 7.1.	Sample Language Model: NEWDATES	7-3
Table 7.2.	Use of Wild Cards to Search a Language Model	7-14
Table 8.1.	Star Phrases	8-12
Table 8.2.	Sharp Phrases and Subsets	8-18
Table 8.3.	Voice Commands and Translations	8-28
Table 8.4.	Help Commands and Translations	8-50
Table 9.1.	Making a Phrase Book Using the <edit> Command	9-3
Table 9.2.	Making a Phrase Book from an ASCII File	9-4
Table 10.1.	Improving Recognition	10-6
Table 12-1.	Improving Recognition Performance by Changing the language model	12-2
Table 12.2.	Index to Edit Functions	12-8
Table 12.3.	Editing Options [^O]	12-9
Table 12.4.	Use of Wild Cards to Search a Language Model	12-12
Table 12.5.	Vocabulary List	12-29
Table 13.1.	Parameters Controlling the Candidate List	13-12

ALPHABETICAL TASK LIST

Use this list to find the instructions for individual tasks. Use the Index at the back of the manual to find more complete task listings and key words.

TASK	Ch/Page
Adding phrases and translations to a language model	12-14
Adding subsets to a language model	12-17
Adjusting the headset	3-2
Adjusting the rejection threshold	11-6
Automating KVS initialization and start-up	14-2
Automating user start-up	14-3
Backing up a phrase book or a language model to floppy disk	4-15
Changing a KVS format language model to ASCII (.prn) format	9-8
Changing phrases and translations in a language model	12-16
Changing a subset name	12-19
Changing token resolution	12-30
Changing tokens per phrase	12-31
Checking computation status	4-11
Checking the microphone	4-5
Choosing vocabulary phrases	8-2
Comparing <edit> and a spreadsheet for making a language model	6-3
Computing a phonetic model on the application vocabulary	13-6
Connecting the KVS to a PC XT or AT	2-4
Copying a phrase book from one computer to another	4-15
Copying translations from one subset into another subset	12-20
Correcting recognition errors by voice command	5-6, 13-11
Creating a batch file	14-5
Dealing with background noise	13-8
Defining KVS output format and destination	13-13
Defining translation strings	8-4
Delaying KVS message transmission	11-9
Deleting a phrase book from disk	12-23
Deleting phrases from a language model	12-15
Designing voice commands	8-13
Dividing a vocabulary into subsets	8-9
Enrolling on an application vocabulary	13-7
Executing several commands with a single phrase	13-10
Experimenting with voice recognition	13-11
Finding problem phrases: using ordered output mode	11-4
Finding how many tokens you can record in training	12-26
Finding how many tokens your phrase book has stored for a given phrase	10-8
Finding how much memory you have for token addition	10-5
Finding when to change a language model	12-2
Finding which subset is active	11-5
Finishing enrollment	4-11
Fixing a bug in DOS 2.0 and 2.1 for the PC XT	C-4
Improving recognition accuracy for selected words: adding, replacing and deleting tokens	10-6
Improving the language model: making phrases more distinctive	13-2
Improving the language model: removing confusions by subsetting	13-3
Improving tokens: changing token resolution	13-4
Improving the enroll phonetic model	13-5
Installing KVS software on a PC XT or AT	2-7
Installing WORDKIT on a PC XT or AT	2-9
Interrupting recording	4-9

Listing available phrase books and language models for editing	12-21
Listing language model contents	12-27
Making a KVS format language model using WORDKIT and <edit>	7-7
Making an ASCII format language model using a text editor	7-25
Making an ASCII format language model using WORDKIT and a spreadsheet program	7-19
Making a new phrase book from an ASCII format language model	9-4
Making a new phrase book from a KVS format language model	12-6
Making a new phrase book from scratch with <edit>	9-2, 12-6
Making a voice profile	4-2, 13-7
Merging two language models to make a new phrase book	12-25
Moving a phrase book between computers	4-15
Naming an application voice profile for the phrase book	12-33
Pausing and resuming token recording	10-5
Pausing between phrases	3-6
Positioning the microphone	3-3
Printing a language model	9-6
Publishing a language model for others to use	12-24
Putting a new phrase book in a user's directory	9-9
Recording tokens for an application	10-3
Recording tokens for enrollment	4-7
Removing a subset and all its translations from a language model	12-18
Resolving communications conflicts	C-2
Resolving compatibility questions	C-1
Resolving keyboard conflicts	C-3
Restarting the KVS	2-15
Resuming token recording	4-10
Saving a phrase book to disk	12-22
Saving disk space and KVS memory when storing phrase books	12-32
Saving enrollment tokens	4-9
Saving trained tokens	10-5
Searching a language model for selected phrases and translations	12-29
Skipping tokens during recording	10-4
Speaking to KVTEST, the KVS application program	5-2
Speeding up entry of digits	13-9
Starting enrollment	4-5
Starting the KVS	2-11
Starting training	10-2
Stopping and resuming recognition	5-6
Storing a language model	14-5
Synchronizing KVS-PC communications	2-15
Transferring a language model between users via the fixed disk	14-6
Transferring a language model between users via the floppy disk	14-7
Troubleshooting enrollment	4-12
Troubleshooting KVS start-up	2-12
Troubleshooting software installation	2-11
Turning off the KVS	2-12
Turning your microphone on and off	5-6
Unpacking and setting up the KVS	2-9
Using existing tokens with a modified language model	10-2, 12-25
Using one phrase to type many translations	5-4
Using star phrases, sharp phrases and sharp subsets	8-8
Using the KVS with FORTE	C-4
Using the KVS with IRMA	C-4
Using the KVS with XTALK	C-3

LIST OF TASKS

NOTES:

FOREWORD

Intended Readers: KVS Users and Vocabulary Developers

The KVS is a speech recognition system that can accurately recognize up to 1,000 user-defined words or phrases. The system includes a desktop unit that connects to a host system (such as an IBM PC XT) and the speech recognition and driver software that resides in the KVS and on the host.

To provide maximum flexibility, the KVS also includes application development tools that allow vocabulary developers to build language models for end-users and programmers to control the KVS from their custom programs.

KVS users can be divided into four groups based on how they interact with the KVS. They are: end-users, vocabulary developers, systems programmers and system managers.

The following table lists the kinds of users, their unique tasks, and the chapters in this manual that contain the task-related information.

KVS User Tasks and Manual Contents By Task

TASK		CHAPTER	
END-USERS	(Speakers who use the KVS with a predeveloped *language model*, a vocabulary list specifying the phrases and translations required to enter data and run their programs by voice.)		
1.	Build a voice profile .	Ch. 3 Ch. 4 Ch. 5	Speaking to your computer ENROLL Try recognition
2.	Record speech samples (*tokens*) by saying the phrases in their language model.	Ch.10	Train a phrase book

KVS User Tasks and Manual Contents by Task

TASK	CHAPTER	
END-USERS (continued)		
3. Save their voice profile, tokens, and language model to disk in KVS format (*phrase book*).	Ch. 10	Train a phrase book
4. Speak to their application programs.	Ch. 11	Test a phrase book
5. Change tokens , as necessary	Ch. 10	Train a phrase book
VOCABULARY DEVELOPERS		
1. Build a language model	Ch. 6 Ch. 7 Ch. 8 App. C	Phrase book creation cycle Tools for making language models Language model design Alphanumeric key codes
2. Convert the language model to KVS format (untrained phrase book)	Ch. 9	Making a phrase book
3. Record tokens	Ch. 10	Train a phrase book
4. Test recognition performance	Ch. 11 Ch. 13	Test a phrase book Throughput: tuning KVS performance
5. Change the language model as necessary	Ch. 10	Train a phrase book
6. Change tokens as necessary	Ch. 10	Train a phrase book
7. Publish the language model for others to use	Ch. 12 Ch. 14	Change a phrase book Distributing Language Models
SYSTEMS PROGRAMMERS		
• Optionally, write C language programs to control the KVS using the KVS Run-Time Library		KVS Reference Manual

KVS User Tasks and Manual Contents by Task

TASK	CHAPTER	
SYSTEM MANAGERS		
• Integrate the KVS into the computing environment	App. C	Hardware and Software Compatibility
• Automate the end-user's boot procedure to include KVS load and set-up commands	Ch. 14	Tailoring end-user Start-up

MORE ABOUT KVS USERS

The typical KVS user is someone who gets a job done by voice. The job may be dictating radiology reports, entering data into a manufacturing data base, inspecting goods for quality assurance, or using data collection systems. The KVS has hundreds of specific uses.

The KVS can be used in two modes. As a transparent keyboard , it is connected between the user and the PC and invisibly converts the user's speech to keystrokes that are forwarded to the program running on the host. This use of the KVS requires no programming. It does require a vocabulary developer to make a language model for the application vocabulary.

The vocabulary developer sets up the vocabulary for a PC application program. The developer builds language models and customizes the user interface so that other users can easily integrate voice into their activities. In some situations, a systems programmer may do the vocabulary developer's job, but a knowledge of programming is not required to build language models or to customize the KVS for end-users.

In the second mode, a voice application program running on the PC can control the KVS. Programs that have knowledge of the KVS can provide a different user interface, direct recognition results to a program running on another computer, and further refine recognition choices before they are used. Use of the KVS in this mode requires programming.

A systems programmer customizes an application to accept voice input by writing programs that interact directly with the KVS. To facilitate this programming, programmers can use the KVS Run-Time Library, a collection of routines callable from a C program that communicate with the KVS driver and the KVS language modeling facilities.

A system manager integrates the voice recognition functions of the KVS into the computing environment. The manager provides files to automate the end-user's start-up procedure.

All users must perform the tasks listed under end-users in the table. End-users train the KVS to recognize them when they speak to an application program. The vocabulary developer, systems programmer, and system manager typically perform additional tasks as outlined in the table.

CHAPTER 1
INTRODUCING THE KURZWEIL VOICESYSTEM

CONTENTS:

- KVS Capabilities
- Host Resources Required
- Vocabulary Capacity
- Inside the KVS
- Setting Up for Speech: A Summary
- The Speech Connection
- Overview of The KVS Installation Procedure

KVS CAPABILITIES

Do you enter data into computers, transcribing written forms into codes?

Do you operate a computer-assisted design system, with your hands on a mouse? Is your visual concentration disrupted by having to move the mouse to a different part of the screen to select commands?

Do you dictate computerized medical records?

Must you interrupt your manufacturing inspections to type or retrieve data at your computer terminal?

Wouldn't it be nice to have a way of communicating with your computer besides typing to it?

The Kurzweil Voicesystem, a 1,000-word speech recognition and response system, makes voice commmunication with a computer practical for the first time. With its large vocabulary and on-board storage of speaker samples (*tokens*), the KVS can be used in many different ways. The KVS boosts productivity in voice-based technical work stations, makes information more easily accessible to decision makers, speeds the process of medical reporting, simplifies data entry in all aspects of computer-integrated manufacturing, and enhances productivity in many other applications.

Your Voicesystem, Kurzweil AI's first in a family of recognition products, is a speaker-dependent, discrete-word recognizer for voice communication of data and commands to an IBM PC XT or PC AT computer. Connected to any IBM or compatible running MS-DOS, the KVS can recognize spoken phrases, translate them into ASCII characters, and send the translations to a program running on the PC.

The KVS can also send recognized spoken words or commands to a program running on a mainframe via a PC. In this configuration, the PC is connected to the mainframe using popular ASCII or IBM 3270 terminal emulators.

Without modifying host hardware or software, the KVS can be used as an alternative to keyboard input or in parallel with it. In this mode the KVS is *transparent* to the application program receiving inputs. Input to the host application appears in MS-DOS as if it had been typed, and the program, already designed for keyboard input, operates as usual, with no sign that the keyboard characters are supplied by the KVS as the result of recognizing speech.

No programming is required to use the KVS in this mode. A user simply develops a language model for the application software and

trains the KVS to his or her voice. In transparent mode, the KVS is controlled by voice command or from the keyboard.

Alternatively, with programming, the KVS can be controlled from a host application program. Voice application programs with knowledge of the KVS can interface the KVS to other computers or devices, customize the user interface, and process words prior to display. The KVS Run-Time Library, a library of routines written in C, supports interactions between the application, the KVS, and the language model. Any program written in C for the IBM PC AT can call subroutines from the KVS Run-Time Library.

For programmers who wish to support the KVS under other computing environments, the KVS Porting Manual describes the MS-DOS implementation of the KVS and outlines the task of porting the KVS driver, Run-Time Library, and host utility programs to another operating system.

HOST RESOURCES REQUIRED

Figure 1-1 shows KVS requirements for disk space and host RAM.

KVS system software uses the same amount of the IBM PC fixed disk no matter how many speakers use the KVS. KVS system software requires about 720 Kbytes of the fixed disk, about 20 Kbytes of host RAM, and one dedicated serial port for communications (COM1 or COM2). If your system has the capacity, you can operate a modem and parallel printer at the same time as the KVS. KVS system software uses no reserved memory address space, which makes continued compatibility likely with personal computer models IBM may release after 1986.

Hardware requirements for the IBM PC and a list of interrupt locations the KVS uses appears in Appendix C.

In addition to the system software requirement, the storage requirement for one person's vocabulary is 250 -- 350 KBytes on your computer's disk memory.

VOCABULARY CAPACITY

The KVS 1,000-word vocabulary can recognize words associated with one program or a cluster of programs used together. For example, your

Figure 1-1 KVS memory use - host PC RAM and disk space used

entire 1,000-word allowance can be spent on Lotus 1-2-3, or half can be used on Lotus and half on DOS. If 500 words is more than you need for any one program, you can include another vocabulary such as WORDSTAR (tm). Combining vocabularies from the programs you use most frequently allows you to change from one program to another without the delay of loading a new vocabulary.

A single speaker can have many 1,000-word vocabularies on disk, but only one vocabulary can be loaded into the KVS. A group of speakers can share the KVS if each uses a smaller vocabulary (for example, if 3 to 10 people each have a vocabulary of from 100 to 300 words).

INSIDE THE KVS

Figure 1-2 shows the desktop KVS next to an IBM PC AT.

The case contains three circuit boards, their interconnecting cables, a power supply and a fan. All KVS hardware is contained in the case, except for the microphone, which plugs into the front panel (Figure 2.1). The power ON/OFF switch is on the rear panel out of the way, since the KVS is generally left running. There are no user-serviceable components inside the KVS unit

The KVS is sturdy, stable, and quiet. No controls are used in normal operation. Speaking level indicators are located on the front and connectors on the back panels. A mode-setting switch is accessible through the base.

SETTING UP FOR SPEECH: A SUMMARY

Most programs that run on the IBM PC XT or AT can be controlled by voice; most programs that take data from the keyboard can take it from a microphone. Each program you wish to control by voice requires a separate KVS vocabulary.

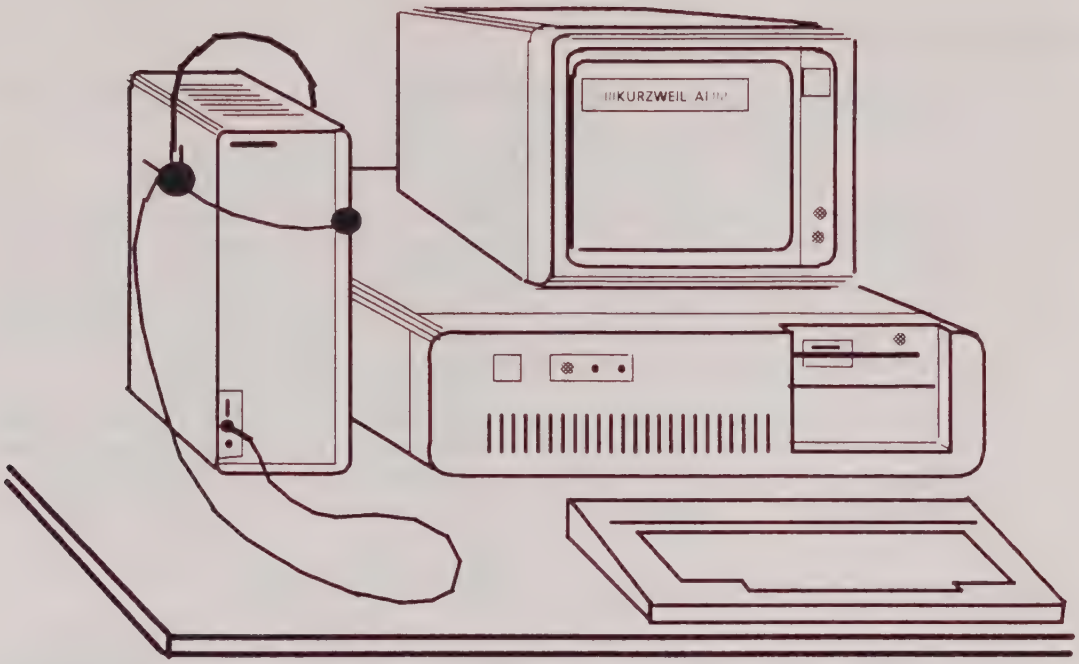

Figure 1-2 Desktop KVS, Headset, and IBM-PC AT/XT
- showing correct position of stored microphone -

The set-up for a recognition task begins with the creation of a list of phrases -- the *vocabulary* -- associated with a program you wish to run by voice. For most programs, developers have already determined how best to use speech recognition. They have chosen words or phrases the KVS will recognize and the translations -- character equivalents -- the KVS will send to the program. The developers have typed this information into the computer and saved the resulting files (called a *language model*) on your computer's fixed disk.

When you first use the KVS, you go through a process call *enrollment.* This produces a *voice profile*, a mathematical model of your voice.

After enrollment, you can set up and train the KVS to recognize vocabularies for any program you want to run by voice. Training creates a personal version of the language model -- a *phrase book* -- including samples of your voice (*reference tokens*) saying each vocabulary phrase. The voice profile you make during enrollment will be part of every phrase book you make after that.

THE SPEECH CONNECTION

When you speak to a program running on the PC, how does the program "hear" what you say?

As you speak into the microphone, the sounds captured by the microphone are digitized in the KVS and compared to stored reference patterns you have made during training. When a match is found, the KVS finds the translation of the match and forwards this translation to the IBM PC through its RS-232 serial communications port. KVS driver software in the host merges this spoken text with your keystrokes and makes the resulting text appear as if typed.

At the same time, the signal from your microphone goes to the earphone on your headset so you can monitor how loudly you are speaking. You will always be able to hear how your voice sounds to the KVS.

OVERVIEW OF THE KVS INSTALLATION PROCEDURE

To install your new KVS, you first connect the communications cable between the KVS hardware and your computer and then install KVS system software and demonstration programs.

Verifying the installation consists of two steps: (1) Enrollment, which makes a phrase book for a special, phonetically balanced vocabulary and (2) an optional test of the recognition accuracy for the enrollment vocabulary.

Chapter 2 tells you how to unpack and connect the KVS hardware, install the KVS system software on the IBM PC fixed disk, and power up your Kurzweil Voicesystem.

- To install KVS hardware and system software, read Chapter 2.
- To power up the KVS and download the KVS system software, read Chapter 2.
- To use the headset microphone, read Chapter 3.
- To record your voice profile (enroll), read Chapter 4.
- To test the recognition accuracy of the enrollment vocabulary, read Chapter 5.
- To develop and train the KVS in your own vocabularies, go on to Part II.

CHAPTER 2
INSTALLATION AND START-UP

CONTENTS:

- Unpacking Instructions
- Setting Up the KVS
- Hardware Installation Instructions and Checklist
- Will the KVS Work with Your PC?
- Installing System Software
- KVS Start-Up: Power ON and KVLOAD
- KVS Power Off
- KVS Restart

NOTE:

If your system is a Kurzweil Voiceterminal (**KVT**), follow the instructions in Appendix D for unpacking and setting up the system. Appendix D replaces this Chapter for the KVT.

UNPACKING INSTRUCTIONS

Your KVS is packed for shipment in a carton with two compartments. One contains the KVS system unit; the other contains accessory parts, this User's Manual, and system software on 3 floppy disks. A fourth disk, WORDKIT, contains useful files for building language models.

Inspect all parts for damage. Inventory the parts and find the following:

- A plastic bag containing 3 cables:

  A KVS power cord

  One RS-232 serial communications cable

  A short adaptor cable for use with the IBM PC AT 9-pin serial port

- A headset microphone
- A binder containing this KVS User's Manual
- KVS system software contained on floppy disks in a sleeve inside the User's Manual
- Wordkit, the KVS demonstration disk

SETTING UP THE KVS

Set up your KVS with the following physical requirements in mind for good performance:

Access

Clear a desktop area of at least 6 x 10" for the KVS. Inside the KVS is a cooling fan vented through the cabinet top. To maintain proper airflow, place the KVS on a hard surface and don't cover the vents. Leave room behind the KVS to make your cable connections.

Power Requirements

The KVS requires a standard 110V grounded outlet and 50 or 60 Hz current. A 220V version is available as a factory option. Conversion to 220V may be done only at the factory or by an authorized repair facility.

Temperature and Humidity

The KVS operates at room temperatures ranging from 50° F to 104° F (10° C to 40° C). The unit may be stored, powered off, at temperatures ranging from 0°C to 50° C. Humidity can range from 0 to 85%, noncondensing, without affecting operation. When the KVS is brought indoors from cold storage, for best results allow 15 minutes warm-up before enrolling or training.

Ambient Noise, Echoes And Small Rooms

The KVS performs best in locations with relatively low noise levels and consistent background noise. Avoid installing the KVS in small rooms with hard surfaces (they produce echoes) and locations where loud and variable noises occur. Consistent background noise will not greatly affect recognition unless it is quite loud, but random loud noises should be avoided if possible. (See also Chapter 13 for tips on recognizing in noisy conditions.)

HARDWARE INSTALLATION INSTRUCTIONS AND CHECKLIST

The installed KVS is pictured in Figure 1-2.

The KVS may be placed as close to the PC as you like, and as far as the RS-232 cable allows. Normally this is 4 feet. Longer cables are available on request.

The microphone cable allows you to be up to 6 feet from the KVS.

The KVS must be within 6 feet of a grounded outlet. Extension cords may be used with the KVS if they are 3-conductor, properly grounded, at least 18 gauge, and comply with UL and local codes. Don't use light-duty home extension cords, two-wire cords, or extension cords with other equipment plugged into them.

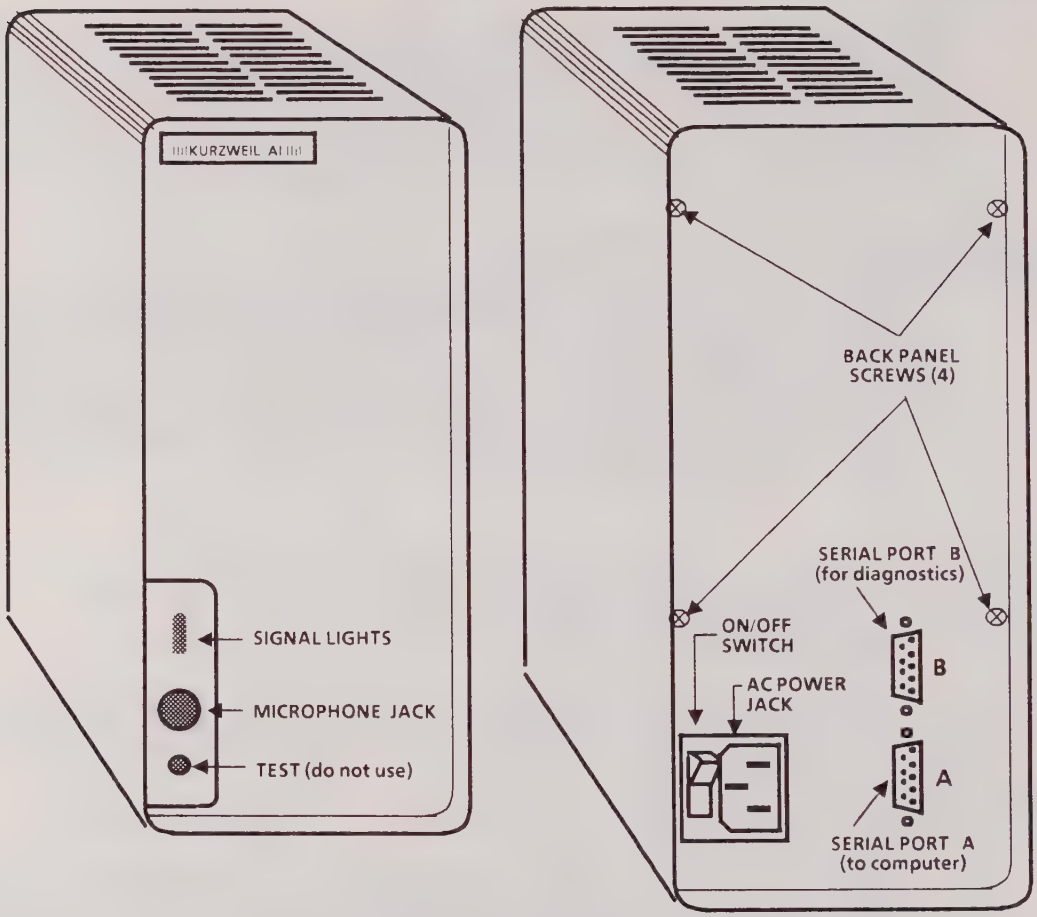

Fig. 2-1 -- KVS Front and Back Panels

Making Cable Connections

With only two cables and a microphone to connect, KVS hardware is simple to install. Use the rear panel reference drawing (Figure 2-2 for a PC XT or similar computers with 25-pin serial communications connectors; Figure 2-3 for a PC AT or similar computers with 9-pin serial communications connectors) to make all your cable connections. Use the panel reference drawing (Figure 2-1) to connect your microphone and locate the signal lights on the front panel. The numbered steps below correspond to the numbers in Figure 2-2 and 2-3.

1. Connect one end of the power cord to the 3-prong power jack at the back of the KVS.

2. Plug the other end of the power cord into a standard 3-hole (grounded) 110V/AC outlet.

3. Plug the small end of the RS-232 connector into port A (the lower 9-pin serial port) at the rear of the KVS.

4. For an IBM PC XT or other computers with 25-pin connectors, plug the large end of the RS-232 connector into the highest numbered serial port at the rear of the IBM PC. Use COM2 if it is free (see below, "Assigning the Serial Communications Port").

5. For an IBM PC AT or other computers with 9-pin connectors, use the adaptor shown in Figure 2-3 to connect the large end of the RS-232 cable to the 9-pin AT serial port.

NOTE: IMPORTANT

Plug the adaptor cable (the short cable) into the computer, never into the KVS. The pins are wired differently and the system will not work if the adaptor is plugged into the KVS.

6. Plug the headset microphone into the 5-pin DIN circular microphone connector at the front of the KVS (Figure 2-1).

7. Be sure to tighten the screws provided on all connectors to avoid breaking connections inadvertently and to maintain radio frequency interference protection.

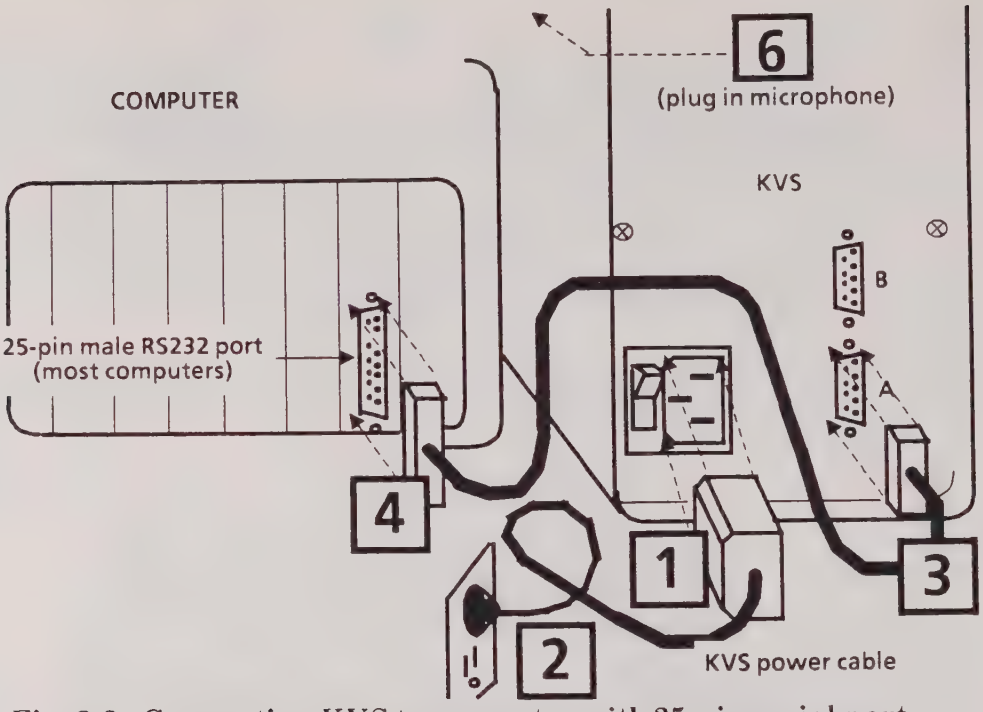

Fig. 2-2 - Connecting KVS to computer with 25-pin serial port

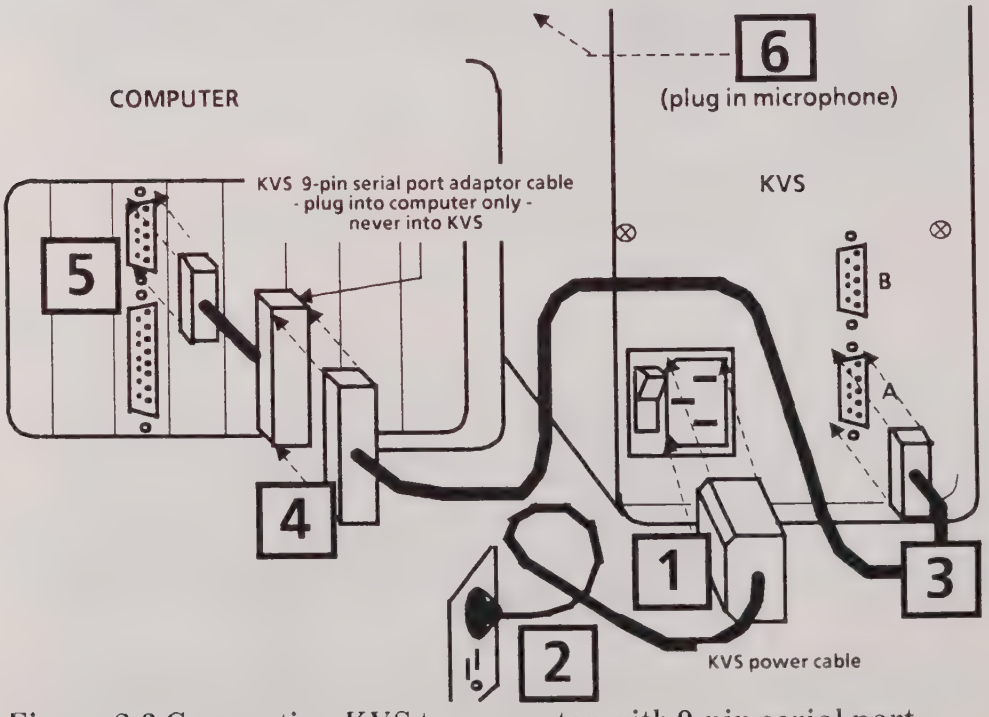

Figure 2-3 Connecting KVS to computer with 9-pin serial port

WILL THE KVS WORK WITH YOUR PC?

The KVS is designed to run with a fixed disk of 10-30 Mbyte capacity. It can also run on a PC that shares a fixed disk through a network, on a two-floppy system with no fixed disk, or a single disk system of 720 Kbytes (3.5 disks). If your PC is a dual floppy system or a single floppy system, please write for the application note "Using the KVS with a 2-disk PC. You can use such a system with the KVS, but you must enroll on a fixed-disk PC.

INSTALLING SYSTEM SOFTWARE

The KVS system software, training software, and demonstration programs are shipped on four 5 $\frac{1}{4}$" floppy disks formatted for an IBM PC . The disks are included in the KVS User's Manual binder.

Once you copy the floppies onto your computer's fixed disk, they are not needed for normal operation. Store them in case you ever need to reload them.

This section gives the procedure for installing KVS system software on a stand-alone PC with a fixed disk. The first part of the procedure installs the KVS operating system from disks 1, 2, and 3; the last part installs the demo disk, WORDKIT.

Instructions assume you have an IBM PC XT or AT (or compatible) computer with power applied.

What INSTALL Does

The INSTALL program puts the KVS driver software and KVS utility programs on your PC's fixed disk and sets up some directories for KVS use. It writes the KVS directory into the path command in AUTOEXEC.BAT and into the device command in CONFIG.SYS if necessary. Before changing CONFIG.SYS, INSTALL renames the original file to CONFIG.OLD and saves it.

When you reboot your computer at the end of installation, the new CONFIG.SYS is read, the new AUTOEXEC.BAT file executes and the correct device and path assignments are made.

Here's how to install the software:

1. Boot your PC and get the MS-DOS C> prompt on the screen.

2. Load the files from the floppy disks:

   (a) Insert the floppy disk labeled KVS System Software, Disk 1 of 3, in drive A. (If there are two floppy disk drives on your computer, use your default drive.)

   (b) Type this command:

A:INSTALL [RET]

where A is your default floppy disk drive. (If you are using drive B, type B: INSTALL [RET].)

(c) The INSTALL program now prompts for a directory pathname. KVS system software will be written into the directory you supply:

"Directory to install programs into (C:\KVPROG):"

Type a directory name or just press [RET]. If you press [RET], KVS system software will be written into C:\KVPROG. Skip to step 2(d) unless you see either of the following messages from INSTALL:

(1) **"Insufficient disk space for installation"**

If you see this message, check available disk space on drive C. You need 700 Kbytes for the KVS operating system. (See "If INSTALL Didn't Work" below.)

(2) **"Your path (C:\;C:\123) does not include your KVS installation directory.** Please add C:\KVPROG to your path specification."

If you see this message, use EDLIN to add the following line to AUTOEXEC.BAT (or the .BAT file that sets your path for working with the KVS):

PATH = C:\KVPROG [RET]

If you supplied a different directory in step 2(c), set the path to that directory instead of C:\KVPROG. (EDLIN commands are summarized in Appendix B.)

(d) When prompted, remove disk 1 and insert disk 2, labeled KVS System Software, Disk 2 of 3.

(e) Press any key. The files on disk 2 will be installed in the directory you have specified.

(f) Repeat for disk 3, when prompted.

3. After you do this, the screen will show the following message:

"DELETING REDUNDANT VOICE PROFILES: This will allow your phrase books to load faster and they will take up less space on your disk. You will not be able to recognize with KVS Version 1.1 software after deleting your redundant voice profiles.
Press [N] if you don't want to delete redundant voice profiles or press any other key to continue."

Redundant copies of the Enroll voice profile will be deleted when you press any key. Press [N] to avoid deleting copies of the ENROLL voice profiles.

4. After you do this, the screen will show the following:

"REMOVE THE INSTALLATION DISK AND REBOOT PLEASE"

5. Remove the floppy disk from the drive.

6. Reboot your computer. For an IBM PC XT or AT, hold down [CTRL] and [ALT] and press [DEL]. (If your computer reboots some other way, do that.)

Installing WORDKIT

Table 2.1 shows the contents of the WORDKIT disk. The disk contains some pretyped language models in the \KVSLMOD directory, useful when you are assembling a new language model with the KVS <edit> command. The disk also contains pretyped vocabulary in the \123files directory. The .wks and .prn files are useful when you are making a new language model with a spreadsheet program. The uppercase files in Table 2.1 are installed automatically by the INSTALL program described in this section.

TABLE 2.1 CONTENTS OF THE WORDKIT DEMO DISK

123FILES <DIR>

FILE NAME	COMMENTS
WORDKIT.WKS	KVS control commands (worksheet format)
YRSLONG.PRN	years of the decade (1980, 1981, etc.) (text file)
MONTHS.PRN	months of the year (text file)
engdig.prn	number words (zero to nine) (text file)
numdig.prn	numerals (0 - 9) (text file)
alphalet.prn	alphabet as letters (a, b, c) (text file)
alphapho.prn	phonetic alphabet (alpha, bravo) (text file)
statesln.prn	states of the USA (text file)
stateabr.prn	states of the USA, abbreviations (text file)
yrsshrt.prn	years of the decade ('80, '81, etc.) (text file)

KVSLMOD <DIR>

WORDKIT.KLM	KVS control commands and numerals
WORDKIT.KMP	

TABLE2.1 (cont.) CONTENTS OF THE WORDKIT DEMO DISK

FILE NAME	COMMENTS
	KVSLMOD <DIR>
DATES.KLM DATES.KMP	months of the year, years of the decade
VOICECMD.KLM VOICECMD.KMP	more KVS control commands
digits.klm digits.kmp	number words and numerals
alphabet.klm alphabet.kmp	alphabet and phonetic alphabet
states.klm states.kmp	states of the USA and abbreviations
INSTALL.EXE	

Use the following procedure to install the demonstration language models (INSTALL will overwrite any file on the fixed disk that has the same name as a file on the floppy disk):

1. Insert the floppy disk labeled WORDKIT Demo Version 2.0 in drive A.

2. Type this command

   A: INSTALL [RET]

   where A is your default floppy disk drive.

3. Type [RET] to install WORDKIT into \123FILES, or type a pathname to install WORDKIT into a different directory when INSTALL prompts for a directory:

   "Directory to install program into (C:\123FILES):"

4. After doing this, the screen will show

   "WORDKIT has been installed. See Chapter 7 of the KVS User's Manual for a tutorial using WORDKIT to build your own language models."

Any files on the WORDKIT disk that are not automatically installed can be copied from floppy to fixed disk with the appropriate DOS COPY command, either

COPY A:\kvslmod\alphabet.* C:\kvslmod*.*[RET]

or

COPY A:\123files\alphapho.prn C:\123files*.* [RET]

Now that the KVS system software and WORDKIT disk are installed, go on to the section called "KVS Start-up: Power ON and KVLOAD," below.

If INSTALL Didn't Work

You may see one of the following messages:

1. "Insufficient disk space for installation"

   The basic KVS operating system takes up about 700 Kbytes of disk space. WORDKIT takes up an additional 100 Kbytes. INSTALL will not complete if disk space is insufficient. If you get this message, examine your directories and delete files to free up at least 700 Kbytes of disk. You will need another 600 Kbytes to create a voice profile (enroll).

2. "Not ready, error reading drive A; Abort, Retry, Ignore?"

   There is no disk in drive A, or the door isn't closed, or the disk is corrupted. To retry press [R]. If you have two floppy drives, try the other one: press [A] for abort, and then type B:INSTALL. If the disk is corrupted, obtain a new one by calling Kurzweil AI, Customer Support.

3. "Bad command or file name"

   The command was typed incorrectly, or the wrong disk is inserted; insert the correct disk and type the INSTALL command again. Install the set of system floppies in label order: disk 1 then disk 2, then disk 3.

4. "Wrong disk!"

   You inserted the wrong disk after INSTALL had begun. Insert the right disk and press any key to continue.

KVS START-UP: POWER ON AND KVLOAD

This procedure assumes that your PC has power, that the KVS software has been installed on the hard disk, and that the microphone is plugged in.

To bring up the KVS:

1. Locate the power ON/OFF rocker switch on the rear panel of the KVS (Figure 2-2) and turn it on. Four events will happen when power is applied and software is downloaded:

   (a) A power-on self-test runs. After turning the power on, wait about 15 seconds for the self-test to complete.

   (b) With sufficient background noise, the green speaking-level light on the front panel of the KVS will come on intermittently, indicating that the KVS is hearing background noise. At this point, the KVS has power but no instructions.

   (c) When you type KVLOAD, there will be a short delay while the loader is read from disk and messages from the loader appear on the screen.

   (d) At the end of the download, the KVS window will pop up.

2. With KVS power on and the self-test complete, at the C>prompt type

   KVLOAD [RET]

   During the 3 minutes while KVS system software is downloaded from the IBM: the screen shows

   "KVS PASSED SELF-TEST: FIRMWARE REV xx
   DOWNLOADING......"

   and the dots (. . .) paint across the screen from left to right. When the line of dots is full, the message changes to

   "SUCCESSFUL DOWNLOAD, please stand by"

   followed by the KVS pop-up window (Figure 2-4). At this point, the KVS is powered up and ready to use.

If the download fails, see "If KVLOAD Didn't Work" below.

KURZWEIL VOICESYSTEM
VERSION 2.0 June 1986
COPYRIGHT 1986 Kurzweil Applied Intelligence

Figure 2-4 KVS pop-up window

If KVLOAD Didn't Work

You may see one of the following messages.

1. "Bad command or file name"

The path command is wrong. Insert the line PATH=C:\KVPROG [RET] in your AUTOEXEC.BAT file, reboot the PC, and type KVLOAD again. Or type the path command and, without rebooting, type KVLOAD again.

2. "KVS not responding on COM1 or COM2, press CTRL-C to cancel"

   The KVS is not plugged in or powered on, or the KVS-PC connection is reversed or wrong. Check the RS-232 connection and the power cord, or turn on the power, wait 15 seconds, and type KVLOAD again, if you stopped the program.

   (a) The RS-232 cable connection is broken. Connect the cable again and tighten the screws.

   (b) The RS-232 port and the port specified in KVPORT=n are different. Type SET KVPORT=n, where "n" is the PC port the cable is plugged into, either 1 or 2.

   (c) The PC lacks the right serial card.

   (d) One of two COM ports is not configured correctly. Plug the KVS into the other port, type SET KVPORT, and try again. Also consult the hardware documentation for the serial card. If your PC has two serial cards, assume that the first card came with your PC and is configured according to IBM specifications and that the second card, COM2, may be a later add-on. Open the system unit and remove the second card. Set KVPORT=1 and run KVLOAD. If the download works, the second card has the problem.

   (e) The unit contains a nonstandard UART communications chip or communications card. The PC serial card dedicated to KVS communications is the standard IBM PC XT or AT RS-232 communications card, or a compatible. A compatible card will work if it has

      1. an 8250 communications chip,

      2. an interrupt request level of 4 (IRQ4) if COM1,

      3. an interrupt request level of 3 (IRQ3) if COM2,

      4. and if it is capable of running at speeds between 9600 baud and 57600 baud.

      If the necessary hardware isn't present, the download fails with the message "Unable to communicate over COM1." In that event, check your hardware documentation to be sure your serial card has the necessary hardware and is set up correctly.

Assigning the Serial Communications Port

One PC serial port must be fully dedicated to KVS communications. The port is logically assigned to the KVS through a DOS command that tells the KVS driver which port to send messages to.

The KVS driver will try to use COM2 for communications unless you assign the port. To assign the port, type (after booting and before download)

Set KVPORT = n

where "n" is the highest available serial port, either 2 or 1. If you are unsure of the port assignment, type SET [RET]. Place the Set KVPORT command in your AUTOEXEC.BAT file for execution at start-up when the PC is booted. (See Chapter 14, "Automatic KVS Download.")

3. "KVS driver not installed"

   (a) You didn't reboot the PC after installing the KVS. Reboot and type KVLOAD again.

   (b) The DEVICE command in CONFIG.SYS is not correct. Change the drive and directory specification in the DEVICE command to reflect the path specification you typed to the INSTALL program:

in file:	CONFIG.SYS
change:	DEVICE =
to:	DEVICE = C:\KVPROG\KV3DR.SYS

   where C:\KVPROG\KV3DR.SYS means the installed location of the KVS driver software.

4. "Unsuccessful download"

   (a) There is a random communications error; try again. Power off the KVS, reboot the PC, and power on the KVS.

   (b) Set the serial line speed lower. At the C > prompt, type

SET KVSPEEDA = 28800 [RET]

or

SET KVSPEEDA = 19200 [RET]

5. "KVS software was already loaded. Do you want to reload (Y/N?)"

   The KVS is already loaded. You may reload or abort the KVLOAD command. A reply of [N] (no) returns your DOS prompt, and you can run any program or type any KVS command. A reply of [Y] (yes) reruns KVLOAD.

6. The green speaking-level light on the front panel of the KVS does not flash. Check the MIC-LINE switch as follows:

   Check that the two-position MIC-LINE switch under the KVS is pushed back toward the rear panel of the KVS. To find the switch, reach in the small square cutout under the KVS (see Figure 2-5). Using a pencil or small screwdriver, push the switch toward the back panel of the KVS. This is the MIC position. The forward position is the LINE position, used only for debugging. With the switch set on MIC, the light will flash.

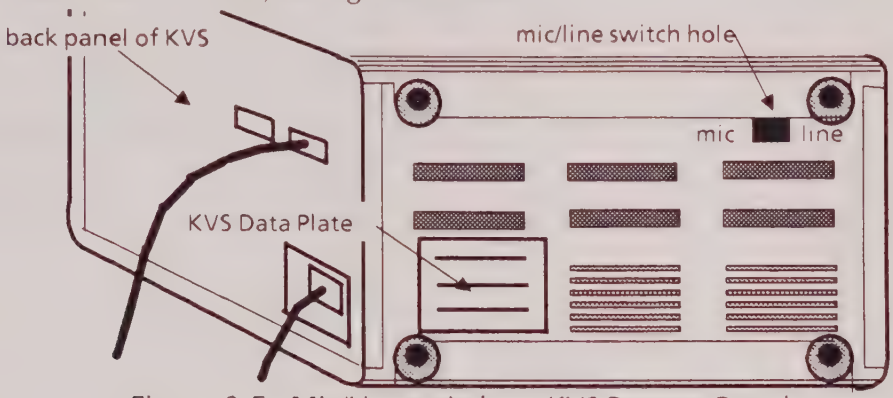

Figure 2-5: Mic/Line switch on KVS Bottom Panel

KVS POWER OFF

Remember to save phrase books to disk before you power off the KVS. Editing or training is lost unless it is saved to disk.

To shut off power, switch the KVS power ON/OFF rocker switch to OFF (see Figure 2-2).

KVS RESTART

There are four situations in which you might want to restart the system. They call for four different strategies:

1. If the PC doesn't work, reboot it or, if necessary, turn the power to the PC off and on again.

2. If the PC does work but the KVS does not respond, abort all KVS processes and start the system software running from the beginning. To do this press [<]. A three-line window containing the [<] will open at the bottom of the screen. Type RESTART [RET].

3. If the KVS is loaded, but the [<] key doesn't cause the window to appear, type KVRESYNC [RET].

4. If the <restart> or KVRESYNC command doesn't work, type KVLOAD at the operating system prompt.

If the KVS does not respond after any of these four strategies, turn power off on the KVS and follow the power-on procedure from the beginning.

CHAPTER 3
SPEAKING TO YOUR COMPUTER

CONTENTS:

- Headset Adjustment
- Microphone Position
- Speaking Tips

The next steps in using the KVS all involve speaking into the microphone. The standard headset microphone supplied with the KVS is a noise-canceling, dynamic microphone that transmits your speech clearly and filters out background noise when you wear it correctly.

Figure 3.1 shows the recommended microphone position. Notice that the microphone is placed close to the lips and near the front of the mouth. Accurate recognition begins with correct microphone placement.

When you enroll on the KVS, you are prompted through a microphone check that lets you adjust the gain and lets the KVS measure the ratio between signal and background noise. During the microphone check, the speaking-level lights on the front of the KVS should be your guide. Your goal is to keep the green signal steady when you talk and make sure the red signal does not come on. If you use the microphone in one specific position for your whole session with the KVS, these adjustments assure good accuracy. The microphone check procedure is explained in Chapter 4.

NOTE:

Whenever you reposition the microphone or take off the headset, you should repeat another microphone check to maintain accuracy.

Take a moment to be sure you are wearing your headset correctly and to think about how you will speak.

HEADSET ADJUSTMENT

Before you put on the headset, look at how it slides, swivels, and rotates. Then follow these directions for putting it on:

1. Adjust the headband as you put it on, to a size that fits your head. The foam-covered earphone should cover your ear like an earmuff, and the rubber-padded clamp at the other end of the headband should grip your hair gently. The band itself goes over the top of your head.

NOTE:

If you have long hair, put your hair behind your ear so that the earphone is directly over your ear.

2. Attach the collar slip to your shirt. This helps bear the weight of the microphone cable and helps keep the microphone position constant. The collar clip slides up and down the microphone cable. Squeeze it to make it slide.

MICROPHONE POSITION

Next refer to Figure 3-2 as you position the microphone. A mirror near your desk will help you. Your goal is to find a consistent and comfortable position. The microphone gives best results when positioned consistently each time you use it. The microphone head is mounted on a light-weight boom held in place by a mounting post attached to the headband. The boom swivels, slides in and out of the mounting post, and moves toward or away from your mouth. It also flips all the way over, to put the earphone over your other ear. To place the microphone, follow these steps:

1. Decide which ear you want to put the earphone over. Flip the boom if necessary. Put the headset on.

2. Swivel the boom so that its curve follows the curve of your cheek.

3. Place both hands on the headset, one hand on the boom, the other hand on the boom mount.

4. Now you can easily slide the boom backward or forward on an axis across your shoulder. Slide the boom until the foam-covered microphone head is at the corner of your mouth and about a finger's thickness away from it.

5. Still holding the boom mount, move the microphone head toward or away from your mouth until it's no more than a finger's thickness (about $\frac{1}{4}$") distant from your lips.

6. Plug the jack end of the microphone cable into the circular connector on the front panel of the KVS (see Figure 2-2).

7. The microphone has a 6-foot cable; to shorten it, gather the slack in the cable tie provided.

Checking Microphone Position

Look in a mirror.

1. Put your finger in front of your mouth as if you were saying "Shhh!" Move it slightly toward the corner of your mouth. This is where your microphone should be.

2. Check it: Is your microphone placed between the corner of your mouth and the center of your lips?

   - Say "boo." Does the microphone clear your lips when you pucker up?

   - Is the microphone no more than $\frac{1}{4}$" (a finger's thickness) away from your mouth?

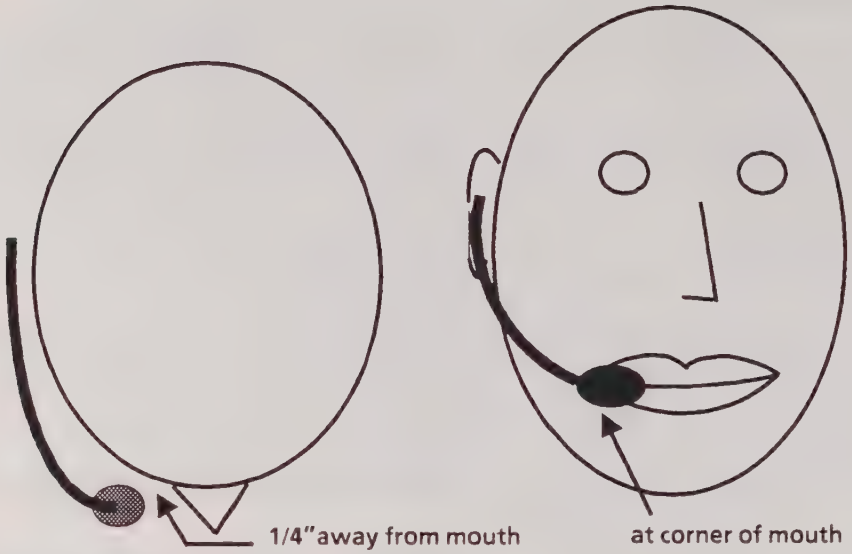

Figure 3-1. Headset Microphone Placement

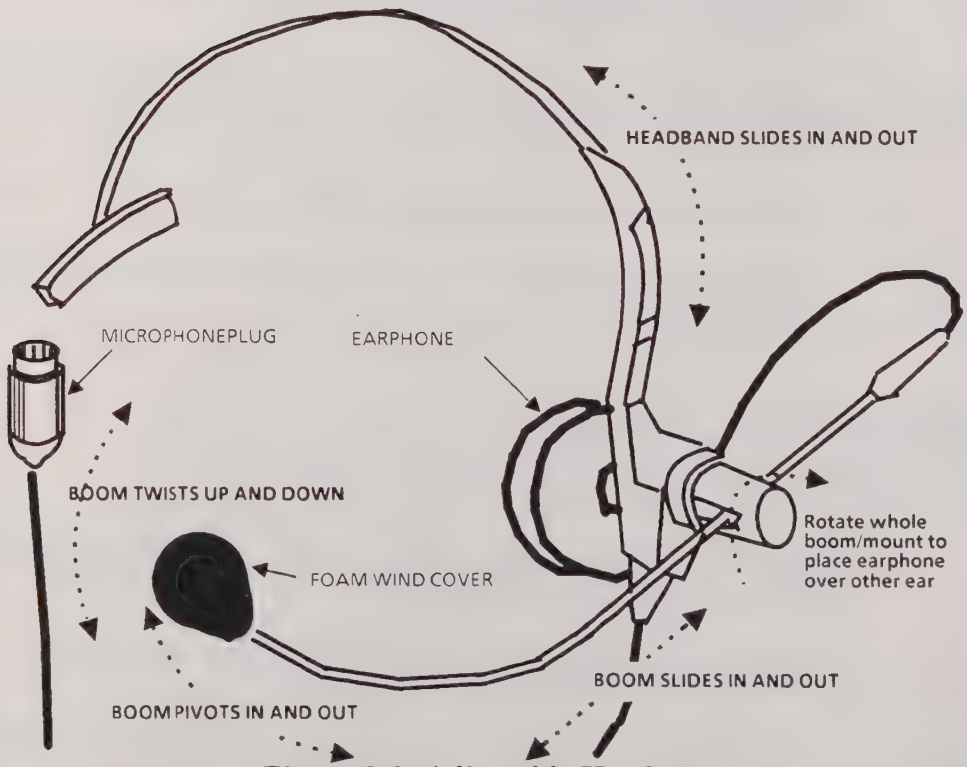
HEADBAND SLIDES IN AND OUT
MICROPHONEPLUG
EARPHONE
BOOM TWISTS UP AND DOWN
FOAM WIND COVER
Rotate whole boom/mount to place earphone over other ear
BOOM SLIDES IN AND OUT
BOOM PIVOTS IN AND OUT

Figure 3-2 Adjustable Headset

Caring for the Headset Microphone

When you take the headset off, lay it down so that the microphone is face up. Hold the microphone by the jack when you unplug it from the KVS socket. Never pull the microphone out of its socket by the cable.

The foam cover over the microphone acts as a windscreen, filtering out noise and moisture. You can feel the flat top and bottom of the microphone through the cover. The cover can be removed for cleaning but should be left on when the microphone is in use.

SPEAKING TIPS

Once you have completed a microphone check, your microphone is live, ready to transmit what you say to the program running on your computer.

How do you talk to a computer? Not much differently than to a person. You should speak clearly, in a commanding manner, in your normal tone of voice. The computer is not hard of hearing or dim-witted; you don't have to shout to make it understand you. On the other hand, just as with people, you shouldn't whisper.

There is one way in which your manner of speaking will be different. Instead of saying sentences as you normally would, you must pause briefly after each word or phrase. It's as if you were to read the present sentence like a list, this way:

It's
as
if
you
were
to
read
the
present
sentence
like
a
list

You must speak each word or phrase with a fresh attack as you would an item on a list. Ordinarily, the KVS requires you to pause about a sixth of a second (180 milliseconds) between words.

When you record tokens for the KVS (as explained in Chapter 4) the display will show you how to speak at a speed of roughly 60 words or phrases a minute; you will find this rate comfortable when you begin, though you may speed up a bit after practice.

Speaking at the rate of about 60 words per minute permits you to make these pauses so the KVS will know where one word ends and the next begins. This results in very high accuracy.

Learning to pause between words is like learning any new skill: you get better with practice, and you won't have to think about it once you've formed the habit.

What Is a Phrase?

A phrase is an item you speak, followed by a sixth of a second (or more) of silence. A phrase can be one ordinary word or it can be a group of words or even a short sentence. As long as it's followed by silence, the KVS treats it as a unit.

When you train the KVS, the screen displays what you are to say next. If several English words are spoken together, without a pause, they are displayed together (Los Angeles, drive B). Remember that when you pause, you are signaling the KVS that one phrase has ended and another is about to begin. The KVS expects to hear all the parts of a phrase spoken continuously without pauses. You can speak each phrase slowly or quickly, but you must pause between them. The longest phrase the KVS can hear is 2 seconds.

Some phrases, like "arc tangent," have slight pauses in the middle. Say these phrases quickly; don't prolong them or the KVS may think they are two words.

The standard duration of pauses between words is a sixth of a second. This can be shortened or lengthened by adjusting a parameter (wordgap) as part of the KVS language model or in the automatic start-up procedure (see Chapters 8, 13, and 14 for more detail).

Speaking Rate

You can't really speak too slowly to the KVS. But if you do slow down, take care not to distort the way you normally speak, as Eliza Dolittle did learning to shed her Cockney accent.

When you speak fast, you may run short words together. You will know if you're speaking too fast because the KVS is likely to merge short words and respond as if they were one word. That longer word probably won't be in the program vocabulary and so won't be recognized.

The KVS does not require you to change anything about the way you normally speak except for your speaking rate. Speaking with a regular rhythm can help while you are trying to learn to pause between words.

Pronunciation

Pronounce words as you always do. When you are creating a voice profile and a phrase book, you will have to repeat vocabulary items. The number of these repetitions may vary. Since your normal speech has some variation, it's good to have some variation in the way you say words during training.

Clarity

Speak clearly.

Don't chew gum when your microphone is on, don't play a radio, and don't chat with your co-workers.

Avoid adding extra sounds to a phrase, such as saying "um" or smacking your lips as you open your mouth to speak. If the words let you, and if it's comfortable, let your lips remain open in the pause between phrases.

Volume

Speak in a crisp, confident manner, as you would in an across-the-desk conversation. Don't shout or whisper. Get in the habit of glancing at the three LED signal lights on the front panel of the KVS.

- The *green* light flickers on and off whenever the microphone is plugged in and the KVS is downloaded. This shows that the KVS is hearing background noise. When you speak, the light will get brighter, and:

- The *yellow* light will occasionally flicker as you speak. If the yellow light never comes on, even when you speak loudly, the signal from your microphone is weak or the MIC-LINE service switch under the KVS is not in the MIC position (Figure 2-7). (See Chapter 2 for instructions on how to set the switch.)

- The *red* light should be off all the time. If the red light comes on when you speak, the signal from your microphone is too strong. Lower your voice, move the microphone away from your mouth, and/or lower the gain. If red comes on during enrollment or training, recording must be done over. (See "The Microphone Check" in Chapter 4.)

You can also monitor the volume of your voice by listening to yourself through the earphone but the lights are a better indication of the correct speaking level. If your voice sounds too soft, speak up a little. Stay with the default gain setting (gain = 5) unless you get no green

signal at all. Then, if speaking louder doesn't help, raise the gain one notch and check your microphone again as described in Chapter 4.

Consistency

When there is more than one way to pronounce a word, decide which way you will say it, and stick with that. For instance, avoid saying "oh" for zero sometimes and "zero" other times. If you want to be able to say it both ways, put both phrases in your language model so you can train the KVS in both versions.

Changes In Your Voice

The KVS is highly tolerant of changes in your voice, but your voice may be unrecongizable to the KVS when you have a cold. If congestion or hoarseness affects recognition, you may have to retrain the KVS to recognize this new voice quality.

The next chapter describes how to enroll on the KVS. Enrolling is a one-time task that sets up your voice profile, a mathematical model of your voice that the KVS uses to recognizes you. At the end of enrollment, you will save your voice profile to disk where it will become part of every application phrase book you use later.

CHAPTER 4
ENROLL: MAKING THE FIRST PHRASE BOOK

CONTENTS:

- Getting Started
- Your Voice Profile
- Making Tokens
- Starting Enrollment: KVENROLL
- "Testing One Two..." The Microphone Check
- Recording Tokens
- Interrupting Recording
- Enrolling in More Than One Sitting
- Finishing Enrollment
- If Enroll Didn't Work
- Phrase Book Files
- Phrase Book Back Up: KVSMOVE

GETTING STARTED

Getting started on the KVS involves a two-step process:

1. Creating a *voice profile,* a record of your voice saying all the phonetic sounds in the English language. This process is called *enrollment.*

2. Creating *tokens,* speech samples stored in the computer of all the vocabulary you will need for your application program. This process is called *training* an application vocabulary.

This chapter describes the enrollment procedure. Chapter 5 shows you how to test the recognition accuracy of the KVS on the enroll vocabulary. Chapter 10 describes training an application vocabulary.

Every first-time user should enroll using the special, balanced vocabulary developed by Kurzweil AI. This vocabulary contains about 200 words that have been selected to cover all the phonetic sounds of English. When you have said all the words in the enroll vocabulary, the KVS will have a record of your voice uttering the whole range of sounds you make when you speak English. (If you speak another language, you should obtain an enrollment vocabulary for that language from the Kurzweil AI distributor in your country.)

YOUR VOICE PROFILE

The product of enrollment is your voice profile, a mathematical model of the acoustic and phonetic characteristics of your voice. The enrollment procedure is generally executed only once and the resultant voice profile is then used throughout the life of the KVS.

The voice profile is ordinarily computed from tokens of the English enroll vocabulary, which covers all the sounds you make as a speaker of English and lets the KVS accurately recognize a large number of words when you speak. It can also be computed from tokens of other sets of words. The voice profile based on the enroll vocabulary serves for every different application vocabulary you will ever ask the KVS to recognize. A voice profile based on your application vocabulary lets the KVS recognize that vocabulary with highest accuracy, but may not be useful for recognizing other vocabularies. (Application-specific voice profiles are discussed in Chapter 13.)

Every user needs a voice profile. While you can use the KVS with someone else's voice profile, recognition accuracy will be highest when you make your own.

Preview of KVENROLL

All the steps in making a voice profile are triggered when you type KVENROLL. Later in this chapter we take you step-by-step through

the KVENROLL procedure. Briefly, the procedure consists of two parts: recording tokens, and KVS computation.

Recording Tokens

After you type KVENROLL, the KVS will load the enroll vocabulary and then ask for a microphone check. It will ask for a phrase and you'll speak it. It will ask for the next phrase, and you'll speak that phrase, and so on. When the KVS says "Done recording all tokens," it will save all your utterances to the host computer's disk, and start the next step.

KVS Computation

The KVS computes for 30 minutes to create your voice profile. When the KVS says, "Computation for phrase book enroll has completed," you will have 6 new files in your directory on disk. Two of them are your voice profile. The other 4 complete the enroll *Phrase Book*, that collection of information the KVS uses to recognize your speech for the enroll vocabulary. At the end of computation you can test the recognition accuracy of the KVS on the enroll vocabulary.

You can complete enrollment with about 25 minutes of speaking plus the 30 minutes for computation. At the end of this one-time process, you will have a voice profile for use in all later training and recognition as well as a phrase book of the enroll vocabulary to test the recognition accuracy of the KVS.

MAKING TOKENS

A good set of reference tokens lets the KVS make good matches during recognition, even if you might pronounce a word differently from time to time. To make good tokens, you only have to pronounce words the way you usually do and speak confidently.

During the enroll procedure, the KVS collects three similar-sounding tokens of each phrase, so you'll be asked to repeat each phrase three times. You should not try for identical pronunciations; variation is a normal part of speech, and recognition accuracy will be better with some variation in the patterns you record now.

More repetitions may be required if any one phrase you speak is *very different* from the others. Should you cough during a phrase, for instance, you'll need to redo that phrase.

When you record, besides teaching the KVS to recognize you, you'll also teach yourself some new habits. You'll become comfortable wearing the headset microphone and will get used to pausing between words. It may be helpful for you to read through the enrollment vocabulary in Table 4-1 before beginning the enroll procedure.

Table 4-1. THE BALANCED VOCABULARY: WORD LIST FOR THE ENROLL PHRASE BOOK

a (as in A,B,C)	dragonbeast	maroon	snitch
a (as in uh)	drooped	meaningless	soften
drop	eager	merit	song
acupuncture	edges	milk	sour
afraid	embellish	morning	sprained
Africa	etch	move	stretchy
agreeable	exact	mudpuddle	sunshine
an	faithful	muffin	swerved
apple	father	music	table
application	ferocity	nervous	talk
apply	film	nevertheless	teabag
appropriate	flowers	nodding	termite
archive	forsook	noise	that
Arthur	fourth	normal	thee
asterisk	fungus	objectify	theorem
atmosphere	gaggle	oblong	therapy
awesome	gearshift	oof	though
bayleaf	geothermal	open	thought
bean	getaway	orange	three
bicycle	ghost	orbit	thwart
birth	Gilbert	other	to
blues	ginger-root	packrat	tubular
boiling	gloom	partial	tune-up
bottom	gorgeous	peeps	tussle
unbutton	gosh	pincushion	twang
candle	gouge	pitched	twelfth
certain	groundhog	plume	twist
chapter	gung-ho	pooches	typhoid
cheerful	guru	primitive	ugly
chestnut	harsh	puckered	umbrella
chewing	hearse	microphone off	untrue
chimes	heaved	pudding	urge
choir	helpless	retrieves	usual
churn	herb	Rhonda	vapor
class	hipboots	rhythm	vouches
clerk	hoodwink	roach	watcher
coinage	hull	rusted	weed
concentration	hutchinson	scrounge	welsh
couch	iceberg	seasonal	white
crash	in	seventh	width
cruder	jersey	shale	wild
cubism	jock	sheepskin	wonderful
curl	jowl	shortcake	woodwork
cushion	judge	should	yards
dabble	junior	shovel	yesterday
deem	keyword	shown	yield
deft	laundry	sleuth	yoyo
dirt	lisp	slipped	Zanzibar
doll	lobe	smashed	zingers
doodad	luck	smokestack	zombie
dozen	measure	snapped	zoomed
#next choice	#delete last word	#try again	#listen to me
new line			

STARTING ENROLLMENT: KVENROLL

To get started, follow the KVS power-on procedure (Chapter 2). At the operating system prompt C>, type

KVENROLL [RET]

Review the help screens KVS now displays if you are a new user. Bypass them if you like by typing CTRL-C.

Next you will see the prompt

"Please enter your KVS username (user):"

Type your name and press [RET], or supply a name consisting of 1 to 8 characters without spaces.

The KVS then loads the enroll vocabulary files for you to record, with this message:

"Copying ENROLL vocabulary files...."

Next the KVS pop-up window appears. The rest of the KVENROLL procedure is conducted through dialogue that appears in the KVS pop-up window at the bottom of your screen. The window says

```
Loading profile data for ENROLL
Please stand by . . . . . .
```

After a brief interval, the microphone check window appears.

"TESTING ONE TWO...": THE MICROPHONE CHECK

The microphone check has three parts:

- Changing gain (volume) (optional)
- Checking background noise level
- Checking the level of your voice

The microphone check lets the KVS know how loud the ambient noise is in the room and how loudly you are speaking. It also tests the level (gain) of your microphone in these conditions. Recording good tokens depends on checking your microphone before starting to speak.

For this first microphone check, start with setting the gain. The microphone check window will read:

Press	[RET]	to begin microphone check	mic OFF
	[ESC]	to escape	
	[G]	to set microphone gain	

Press [G] to check gain. The window will show the following:

Repeat this sentence, pausing briefly after each word, until you	mic ON
like the volume: "TESTING ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT"	
Press [U] to boost volume, [D] to lower volume, or [ret] to continue	gain = 5

1. Say the test sentence, listening to the microphone signal through your earphone. You can raise or lower the volume (gain) with the U and D keys. This adjusts the KVS internal volume control, as well as the volume in your earphone.

   As you speak, watch the speaking-level lights. *Green* should light when you speak, and *yellow* may flash. If *red* also lights, lower the gain. The gain level is shown at the lower right corner of the window. It starts at 5 and can go up to 9 or down to 0. Moderate gain (gain = 5) is usually good for recognition.

2. Now set the background noise and speech levels. Press [RET] to see this window:

Checking Signal Noise ratio...	mic OFF
When you are ready to be silent, press [RET] and don't speak	
press [ESC] to start over	

Press [RET] to let the KVS measure ambient noise. When you see the next message, don't talk and make sure no one around you is talking. Remain silent until you see the next screen.

Now, when you press [RET], you'll be prompted for the test sentence again. Follow the screen prompt:

Listening to Background Noise...	mic ON

Ready to listen to speech When you are ready to speak, press [RET]	mic OFF press [ESC] to start over

Say the sentence below and pause between words "testing one two three four five six seven eight"	mic ON

The KVS capitalizes each word after you say it. Pause briefly after each word and wait for it to go to uppercase. After you say the last word of the test sentence, the KVS displays the message: "All done. Thank you." and switches immediately to recording tokens (see below).

If the difference between signal and noise is not great enough to insure good accuracy, the KVS may ask you to repeat the microphone check. Prompts on the screen will tell you what to do.

RECORDING TOKENS

KVS recognition accuracy will reflect the care you take in recording tokens during enrollment. To record good tokens, do the following:

- Preview the list of words you will be saying.
- When the KVS prompts you to say a word, speak clearly.
- Speak when text is displayed in the window, not before.
- Don't speak too quietly or too loudly.
- Pronounce words naturally, as you would in conversation.
- If you make a mistake, or cough, or if there is inadvertent background noise such as a phone ringing, back up by pressing [B] and say the word again.
- If you miss a prompt, back up (press [B]) and re-say the word when you are prompted to. Do this when you say the wrong word or the

screen skips ahead. WHEN IN DOUBT, PRESS [B].

Remember that enrollment is a data collection process: take care that the tokens collected are good.

The window for recording tokens looks like this:

```
                                                    mic OFF
1 SAY:   A (as in A,B,C)          [P]    Pause
2        DROP                     [ESC]  Menu
```

The screen displays a short list of phrases: the phrase you must say now and the next phrase to speak.

1. Speak the phrase following the command SAY and then be silent.

2. The screen blinks off and the microphone turns off momentarily.

3. Remain silent for about a quarter of a second until the screen blinks on again, showing you three words: the word you just said, the one to say now, and the next word.

```
1            A (as in A,B,C)     [B]   go back          mic ON
2       SAY: DROP                [P]   Pause
3            MEASURE             [ESC] Menu
```

4. When you see the next phrase marked SAY, speak the phrase. The word to SAY is always on the middle line.

5. Do not say the # (pound sign) before phrases such as *#listen to me*, and *#delete last word*. Just say "listen to me" or "delete last word."

6. Repeat the recording procedure until the top line of the window says "All done." This means you have recorded all the tokens for the vocabulary.

Notice that you can press [B] to back up and say the previous phrase again, or press [P] to pause. [P] lets you take a break of any length. [B] keeps going. You will be prompted for a new microphone check when you resume after pausing. Redo the microphone check if any of the following occurs:

- You change the position of the microphone.
- The level of ambient noise around you changes , as it does for example when the ventilation system turns on or off,

- You take off your headset and put it on again.
- You take a break from enrollment and resume recording.

INTERRUPTING RECORDING

You can interrupt the recording process in several different ways:

1. Pause briefly [P].
2. Back up to replace a bad token [B].
3. Escape [ESC] to the enroll menu and then save [S] your tokens recorded so far, or exit [RET].

Pausing Briefly

If you simply want to pause and take a break, press [P]. The pause screen tells you both the number and the percent of tokens you have recorded so far and how many there are in all. As soon as you press [RET], recording will resume, with a request for a microphone check first. Do the microphone check if you have removed the headset or changed the microphone position. Otherwise, press [ESC] to bypass it.

Replacing A Questionable Token

The KVS records every sound it hears while the SAY prompt is on the screen. If you say something other than the phrase being prompted or if some background noise such as the telephone intrudes, backup [B] and resay the phrase. Your tokens must not contain extraneous sounds.

To back up and erase a token, press [B]. The list of phrases on the screen scrolls down so that the previous phrase becomes the current phrase. Press [B] repeatedly, if necessary, until the phrase you want to redo is marked SAY. When you speak this phrase, the new token replaces the old one.

You can go back as far as you like by pressing [B] any number of times. Since enrollment continues from where you backed up, you must repeat all the words over which you backed up.

Saving Your Tokens

Another reason to pause is to save your tokens to disk. To do this, press [P] to pause, [ESC] for menu, and then [S] to save. If you are concerned about the continuity of your electrical power, use this

command to guard against losing data. Saving tokens during enrollment takes about 10 seconds. When the save is complete, the enroll menu returns:

```
ENROLLMENT MENU: Recorded 5 of 609 tokens                mic OFF

[A] Add tokens          [S] Save tokens          [RET] exit
```

Press [A] to continue recording tokens.

ENROLLING IN MORE THAN ONE SITTING

To break up your enrollment procedure into several short sessions, you can stop partway through recording by pressing [ESC] on the recording window. The [ESC] key returns to the ENROLL menu, showing how many tokens you have recorded so far. If you press [RET] on the ENROLL menu, the tokens recorded so far are saved automatically. At the end of saving tokens, the KVS will show a status message for completion:

```
User {your name}          Phrase book ENROLL          Subset NONE
Current Command action is : enrollment incomplete
```

Resuming an Interrupted Enrollment

When you want to resume enrolling, at the C> prompt, type

KVENROLL {your name} [RET]

and you will see this message:

"Profiling for {your name} will continue from previous session."

Next the KVS window appears, confirming that you are signed on and that data for ENROLL are being loaded. Next the microphone check window is offered. Complete the microphone check to get the ENROLL recording window. Recording resumes where you left off. The prompt will show you the last phrase you said and the phrase to say now.

FINISHING ENROLLMENT

When you have recorded all 609 tokens, you are finished with your part of the enrollment procedure. Now the KVS will take 30 minutes to compute your voice profile. You will see this message:

```
ENROLLMENT MENU. DONE RECORDING ALL 609 TOKENS                mic OFF
```

The KVS will tell you to stand by and then will enter its computation phase with this message:

```
Saving Phrasebook C :\KvSUSERS\ {your name}\ENROLL to disk....done
Beginning Computation phase 1 {and later 2}
Computation will take 30 minutes
```

Now is a good time to take a break. You can use MS-DOS during this time although it is not recommended. Response times will be slow because the PC is performing disk operation for the KVS.

When the ENROLL computation is done, the KVS sends a beep to your terminal and tells you enrollment is complete:

```
User {your name} Phrase Book "ENROLL"                    Subset none
Type <HELP or press any key to return to your operating system.
Current command action is: Enrollment complete
```

Checking Computation Status

During the computation phase of enrollment, you can issue the <!status> command. This command interrupts computation, reads and reports the duration of computation so far with this message:

```
User {your name}  Phrase book Enroll
Current command action is: Voice Profile Computation
Computation phase (1 or 2) started xx minutes and yy seconds ago
```

IF ENROLL DIDN'T WORK

Sometime during the enroll procedure, the following messages may appear or the following conditions may occur. We give you some suggestions for possible causes and remedies.

1. "Sorry you must have at least 600,000 bytes available on your default disk" (after typing KVENROLL).

   (a) The fixed disk (C:) doesn't have 600,000 bytes available. To free up space, use KVSMOVE to back up phrase book files to a floppy disk, delete these files from the fixed disk, and try KVENROLL again. To determine which files to delete, see Table 4.2. Do not delete the ENROLL.KAM, ENROLL.KPM and ENROLL.KPL files. These files constitute your voice profile. After you try recognizing the enroll vocabulary, the ENROLL.KVC file is not useful and can be deleted

   (b) The fixed disk is not assigned as the default drive. To assign the fixed disk as the default drive, type <drive c> and give the KVENROLL command again.

2. "Not a legal user name" (after typing KVENROLL).

   User names are spelled with 1 to 8 characters and no spaces. If you know you entered a legal name, reboot your PC and run KVLOAD again. If the name you entered is not legal, type in a legal user name.

3. The screen doesn't blink off during recording -- nothing happens when you speak.

   For short words like "up" or low-energy words, like "button," say the word again more slowly. Exaggerate the vowels: say "uuup," and "buuttonnn."

4. The screen advances to the next word, but you didn't say anything.

   The microphone picked up noise. During recording, the KVS interprets any sound as the way that you say the current phrase. Back up [B] and say the current phrase again.

5. The screen blinks off before you finish speaking during recording.

   The KVS heard enough silence to think you had finished speaking. The phrase has an embedded pause as in NEVERTHELESS or GINGER-ROOT or APPROPRIATE or you paused for one-sixth second between parts of a multiword phrase. Back up [B] until the last phrase you said is marked SAY. Speak the phrase more quickly and continue recording from there.

6. The screen fails to blink on again after you just said a phrase.

(a) You're speaking too soon. Wait without speaking until the screen blinks on. Back up [B] until the last phrase you said is marked SAY. Speak the phrase. As you record, wait until you see the word marked SAY before you say it.

(b) A loud background noise has caused the KVS to wait for silence. It may be necessary to pause [P] and perform another microphone check.

6. The KVS asks you to SAY a word you just said.

   The KVS consolidated two phrases into one. You said two successive phraes without pausing between them. Back up [B] until SAY points to the earlier phrase of the pair. Speak the phrase.

PHRASE BOOK FILES

At the end of enrollment, there are six new files (about 250 KBytes) in your directory under \KVSUSERS. Table 4.2 shows the files resulting from enrollment and tells which ones must be kept in your directory to do more work.

Table 4.2. ENROLL PHRASE BOOK FILES

FILE NAME	SIZE	PHRASE BOOK PART	FUNCTION	DELETE FILE?
nroll.KAM	30KB	Voice Profile	Train and recognize	no
nroll.KPM	70KB	Voice Profile	a new vocabulary	no
nroll.KPL	1KB	--	Enroll log file	no
nroll.KLM	10KB	Language Model	Recognize the ENROLL	ok
nroll.KMP	5KB	Language Model	vocabulary	ok
nroll.KVC	131KB	Tokens	Recognize the ENROLL vocabulary	ok

NOTE: WARNING

Do not delete the first three enroll files, those with extensions .KAM, .KPM and KPL. They are needed for all subsequent vocabulary training as well as recognition.

The last three enroll files, those with extensions .KLM, .KMP, and .KVC are needed only to recognize the enroll vocabulary. They may be backed up to floppy disk and deleted from the fixed disk when KVS is recognizing a different vocabulary.

PHRASE BOOK BACKUP: KVSMOVE

To free up space on the fixed disk, use the KVSMOVE program to back up the six files that make up your enroll phrase book. Follow these directions:

1. Format a blank floppy disk (360 Kbytes or 1.2 Mbytes).

2. At the C > prompt type

KVSMOVE BACKUP {your name} enroll [RET]

KVSMOVE prompts you to insert a blank formatted floppy disk in drive A and press any key when you are ready. When you do, KVSMOVE warns that any files on your floppy disk will be overwritten, and asks you to press a key to confirm your back up. As soon as you press a key, the KVS copies the phrase book from fixed disk to floppy. It asks you to insert a second floppy if your phrase book doesn't fit on one.

You can restore this phrase book to the fixed disk on another computer or on your own computer if it was deleted:

At the C > prompt, type

KVSMOVE RESTORE {your name} enroll [RET]

and insert the floppy disks when KVSMOVE prompts you to do so.

Once the ENROLL phrase book is backed up on floppy disk, you can delete three of its files from the fixed disk with the DOS DELETE command. Give the DELETE command once for each file to be deleted. At the C > prompt, type

```
DEL \KVSUSERS\{your name}enroll.KLM
DEL \KVSUSERS\{your name}enroll.KMP
DEL \KVSUSERS\{your name}enroll.KVC
```

The KVSMOVE command with the modifier \L can also be used to back up and restore the language model for a phrase book rather than the whole phrase book. To back up the ENROLL language model, type

KVSMOVE BACKUP ENROLL \L [RET]

This procedure is explained in Chapter 14.

If KVSMOVE Didn't Work

If you see the prompt

"Bad command or Filename"

either the DOS BACKUP command or COMMAND.COM is missing from the search path. Be sure that the directory containing the MS-DOS command set is on the path. (Use the PATH command to examine and change the search path.)

Using DOS COPY

KVSMOVE is preferable to DOS COPY for backing up the phrase book files because it knows which files to back up and restore. COPY backs up only what you tell it to. You can, however, use the DOS COPY command instead of KVSMOVE if you prefer. (When restoring a floppy to the fixed disk, COPY into C:\KVSUSERS\{your name}.)

IMPORTANT

Use COPY in both directions, or KVSMOVE in both directions. Don't use COPY from a disk that was created using KVSMOVE/BACKUP or RESTORE from a disk that was made using COPY.

CHAPTER 5
TRY RECOGNITION

CONTENTS:

- Testing Recognition
- How to Use One Phrase to Type Many Translations
- Voice Correction
- Stopping and Resuming Recognition
- Making a Phrase Book for Your Application

Once the computation for enrollment is complete, you can use the KVS to recognize the words and phrases of the ENROLL vocabulary. This chapter describes how to

- test recognition -- start and restart recognition, take a break, and quit;
- use one phrase to type many translations;
- tell the KVS to stop listening;
- tell the KVS to resume listening;
- tell the KVS to correct a recognition error.

Instructions assume the KVS is powered up and the system is downloaded as described in Chapter 2.

TESTING RECOGNITION

The procedure for starting recognition involves a few simple steps:

1. Start up an application program.
2. Sign on as a KVS user.
3. Start recognition.
4. Do a microphone check.

Start up an Application Program

KVTEST is the application program supplied by Kurzweil AI to receive and display recognition results. It is a useful program to practice on while you're getting familiar with the KVS, before you're ready to run the KVS with your own application. You can say the ENROLL vocabulary to the operating system as well, but it's nicer to say it to the blank screen that KVTEST provides. To start the program, at the C> prompt type

KVTEST [RET]

To stop the program, type CTRL-Z.

When you say phrases from the ENROLL vocabulary, KVTEST displays them on the screen just as if you had typed them from the keyboard. KVTEST does not interpret what you say; it does not execute any commands or programs. It simply tests how well your tokens match your spoken phrases.

What can you say to KVTEST? You can say any phrase in the phrase book you download and be recognized. The translations of phrases that

score above a rejection threshold get sent to KVTEST and are displayed on the screen.

Sign On as a KVS User: <user>

To sign on, type your user name, like this:

<user ray>

When you get the confirmation

Ray is signed on

your next step is to ask the KVS to listen to you.

Start Recognition: <recognize>

Type <recognize> followed by a space and the name of a phrase book to download, such as

<recognize enroll >

Of course, you can type any other phrase book name in place of ENROLL. The screen prompts that there will be a short delay while the phrase book you named is downloaded from fixed disk to KVS memory:

```
LOADING PHRASE BOOK enroll
Please stand by . . . . . .
```

When downloading is complete, the KVS prompts you to check the microphone.

```
Press    [RET]  to begin microphone check          mic OFF
         [ESC]  to escape
         [G]    to set microphone gain
```

You went through the microphone check at the beginning of enrollment. In recognition, the procedure is the same. Press [RET] to perform the microphone check. Do not press [ESC] when you begin recognition or you may get poor accuracy. In general, you should recognize with the same gain level you trained on.

As soon as the microphone check is over, the KVS is ready to recognize your speech:

Now when you say a word, its translation appears on the screen. Say any of the words in Table 4.1. The KVS window will disappear as soon as you speak or press a key. At this point you are recognizing and running KVTEST.

HOW TO USE ONE PHRASE TO TYPE MANY TRANSLATIONS

The KVS does not merely type what you say. It can also translate. Each phrase in the ENROLL vocabulary has four different translations, and each set of translations is called a *subset.* The translation subsets in ENROLL are named as follows:

- Subset *upper*: types all phrases in uppercase.
- Subset *lower*: types all phrases in lowercase,
- Subset *living*: types only those phrases associated with living things.
- Subset *define_living*: types definitions of phrases associated with living things.

With four translation subsets, a single phrase such as FLOWERS can cause four different things to be typed. Try this:

1. Type <subset>.

   The KVS displays a window identifying the current subset, for example:

   The current subset is upper.

2. Type <subset lower>.

3. Say "flowers nevertheless." The KVS types

   flowers nevertheless

   because it is using lowercase translations.

4. Type <subset upper>.

5. Say "flowers nevertheless." The KVS types

   FLOWERS NEVERTHELESS

   because it is using uppercase translations.

Whatever translation is listed in the current subset will be typed. If there is no translation for the word in the current subset, then nothing will be typed.

The ENROLL vocabulary has a subset of 25 words that describe living things. When you change to the LIVING or DEFINE_LIVING subset, the KVS will type a response only when one of those 25 words is spoken. If you say another word or make a noise, the KVS beeps and displays a message in the window. Try it:

1. Type <subset living>.

2 Say "flowers." The KVS types flowers.

3. Say "nevertheless." The KVS beeps and displays the message

   That word is not active in the current subset.

4. Say "San Francisco California." The KVS beeps and displays "Please Repeat" because that phrase is not in the ENROLL vocabulary.

A translation can be much longer than the phrase that triggers it. To see how that works, do the following:

1. Type <subset define_living>.

2. Say "flowers." The KVS types a dictionary definition of flowers.

Table 5.1 shows all phrases naming living things in the ENROLL vocabulary. These phrases are the only ones that will be recognized when you switch to subset LIVING or DEFINE_LIVING. You can ask for definitions of any of the words in Table 5.1.

TABLE 5.1 Living Things in the ENROLL Vocabulary

herb	apple	father	termite
flowers	fungus	bayleaf	bean
orange	ghost	gilbert	ginger-root
packrat	groundhog	pooches	chestnut
guru	choir	rhonda	clerk
roach	watcher	sleuth	zombie
dragonbeast			

VOICE CORRECTION

The KVS recognizes three special phrases for voice correction. When the KVS types the wrong word or you say the wrong word, speaking one of these phrases will correct the mistake:

1. "Delete last word"

   The KVS erases the last word typed; you can erase, one by one, the last 10 words on the screen by repeating this phrase up to 10 times.

2. "Try again"

   The KVS erases the last word typed and then listens for you to say the word again. When you do, the KVS will probably get it right. It will not make the same mistake again because it excludes the word just erased as a recognition candidate.

3. "Next choice"

   The KVS erases the last word typed and types its second guess. The second guess is usually the right word.

Try these:

1. Say "flowers" four times. The KVS types

   flowers flowers flowers flowers

2. Say "delete last word" four times. The KVS erases all four words.

3. Say "flowers." The KVS types "flowers."

4. Say "next choice."

   The KVS erases "flowers" and takes a second guess or leaves it erased if has no second guess. If it has no second guess, say the word again; it has your word, but the tokens didn't match well.

STOPPING AND RESUMING RECOGNITION

While you're talking to the KVS, you may want to cough or talk with someone or take a short break or sign off. Table 5.2 shows all the ways to turn off and restart the KVS by voice command and by typing.

Table 5.2 Turning Off and Resuming Recognition

TURN OFF THE MICROPHONE	TURN ON THE MICROPHONE
type <pause> type < say "microphone off"	type <recognize .> type [backspace] to delete the < say "listen to me"

Saying "microphone off" does not really turn the microphone off or even turn recognition off. In fact, the microphone is always on when power is on. This voice command tells the KVS not to begin typing until it hears one special phrase: "listen to me." Try it:

1. Say "microphone off." The KVS window says "Microphone is off. Say 'listen to me' to resume recognition."
2. Say "flowers nevertheless." There will be no response.
3. Say "listen to me." The KVS window says "I'm listening."
4. Say "flowers nevertheless." The translations are typed on the screen.

Pause and Sign Off: <pause>, <bye>

Typing [<] or <pause> also turns off the microphone.

- When you type [<], the pop-up window appears and the microphone turns off. When you press [backspace] or [RET] or a close angle bracket [>], the window goes away, the microphone turns on, and recognition restarts. (See also Appendix B: <parameter listen ON/OFF>.)
- Typing <pause> gives you additional choices, discussed below under "Restarting Recognition."
- Typing <bye> signs you off completely.

Restarting Recognition after <pause>

Pressing [RET], as the <pause> screen shows, restarts recognition. The <pause> command gives some other choices:

KVS is paused			mic OFF
Type	[B]	for bye (I quit work)	[S] for stand by (I'll be back)
	[RET]	for resume	[M] for microphone check, then resume

If you want to perform a microphone check before restarting, press [M]. (See "Repeating or Bypassing the Microphone Check," below.)

You can take a longer break and resume later by pressing [S]. The KVS exits recognition mode. If you had KVTEST running, return to DOS by pressing CTRL-Z. Later, when you resume work, you can type <recognize .> without having to sign on or download your phrase book again. The period in the angle brackets tells the KVS to use the currently loaded phrase book.

Finally, you can sign off after pausing by pressing [B]. The result is the same as typing <bye>. This is an orderly exit from the recognition process. It prompts you to save tokens if you have changed a phrase book and have not yet saved the changes to disk. Changing a phrase book is explained in Chapters 10 and 12. You cannot use your application while the <pause> menu is shown. [B], [S], and [RET] all return you to your application.

Repeating or Bypassing the Microphone Check

On returning to recognition, recheck your microphone if you took your headset off or moved the microphone while you paused.

Be sure your microphone is plugged in and check its placement. Is it between the center and the corner of your lips, not more than $\frac{1}{4}$" away? Is it distant enough so your lips don't touch it when you speak?

You can always initiate a microphone check by typing <pause>. Do this whenever

- recognition accuracy falls off,
- the level of background noise in the room changes,
- you see the *green* speaking-level light dim or go out (speech too soft), or
- the *red* speaking-level light comes on (speech too loud).

Should *green* go out or *red* come on, try adjusting the microphone gain. If *red* comes on, turn the gain down. See also Chapter 13, "How to Minimize Effects of Background Noise." Type <pause>, then [M] to select the microphone check, and [G] to select the gain test.

The system will not let you begin recognition until you check the microphone. You will also be asked to repeat the microphone check if any of the following occurs during the procedure:

- You speak when the KVS expects to hear silence.
- You are silent when the KVS expects speech.
- Your microphone is unplugged.

- You press [RET] twice in succession.

You can bypass another microphone check by pressing [ESC] instead of [RET]. Do this only if you haven't changed the microphone position since the last valid microphone check.

MAKING A PHRASE BOOK FOR YOUR APPLICATION

Now that you have enrolled and tested recognition, you can work on a vocabulary for your own application. If you are designing the vocabulary, go on to Part II of this manual.

If you are an end-user working with a vocabulary developed for you, you are ready to train and test that vocabulary. All the events of training are triggered by typing, at the C> prompt,

KVTRAIN [RET]

KVTRAIN prompts for your phrase book name and your user name. Alternatively, you can start training by typing

<train {phrase book name}>

You can type <train ?> and see all the phrase books available for training. To train and test your own vocabulary turn to Chapter 10, "Recording Tokens."

CHAPTER 6
OVERVIEW OF PHRASE BOOK CREATION AND MODIFICATION

CONTENTS:

- What Is a Phrase Book?
- What Is a Language Model?
- Three Ways to Make a Language Model
- The <edit> Command
- A Spreadsheet Program
- A Text Editor

WHAT IS A PHRASE BOOK?

A phrase book, diagrammed in Figure 6.1, is a collection of information the KVS uses to recognize a user's speech. The phrase book is stored in each user's private directory and is downloaded to the KVS for recognition. Typically, every user has a stored phrase book for each application (or cluster of applications) he or she runs by voice. The phrase book contains both written information (a list of vocabulary phrases and their translations) and spoken information (reference tokens and a voice profile).

Before a user can speak to an application, a developer must build the language model component of a phrase book. As vocabulary developer, your job is to turn the vocabulary list and translations into language model form, the form to be downloaded into the KVS. This chapter previews the cycle of making, training, testing and changing a language model, which is the subject of the next six chapters.

WHAT IS A LANGUAGE MODEL?

A language model functions as a lookup table for the KVS during recognition. It lists all the vocabulary items (words or phrases) that a user can say in the course of running an application. For each spoken phrase (input), it specifies the translation string (output) that the KVS will send to the host computer when it has recognized that phrase. For example, when you say "Lotus" the translation could be the commands to load and start 1-2-3.

The language model you design becomes part of each end-users phrase book together with the user's speech samples from training and enrollment (tokens and voice profile).

You can make a language model in KVS format using the <edit> command or in ASCII text format using a spreadsheet program or a text editor if you don't have a spreadsheet program.

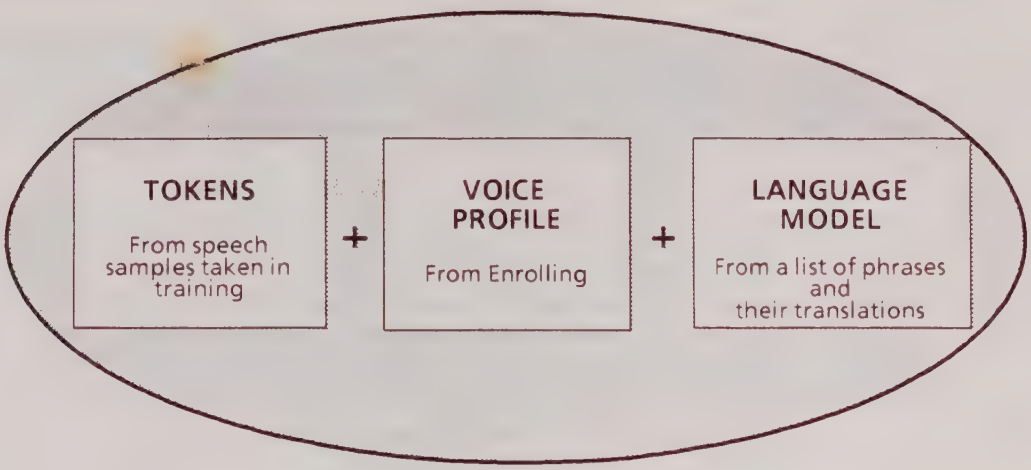

FIGURE 6.1 -- A Phrasebook, with its three parts

THREE WAYS TO MAKE A LANGUAGE MODEL

Making a language model is straightforward as soon as you have chosen a tool to use. The KVS can work with three alternative tools:

1. the KVS <edit> command;
2. any spreadsheet program;
3. any text editor.

You may find yourself using more than one tool over time, depending on the size of the language model, whether you have preexisting word list files in ASCII that you want to include in the language model, and whether the completed language model needs extensive changes.

<edit> does not produce an ASCII file. Instead it creates a language model in KVS format in the user's private directory, ready for training. If you use a spreadsheet program or a text editor, an ASCII file is generated that is then compressed and combined with other files to make a phrase book.

THE <EDIT> COMMAND

<edit> is the KVS on-board language model editing command. This means it requires a connected KVS. You do not need to own a spreadsheet program or a text editor to build language models. To use <edit>, you name the phrase book to contain the language model and type in vocabulary items and their corresponding translation strings one at a time.

Figure 6.2 shows the cycle of phrase book creation and modification using <edit>. Notice that as soon as you invoke <edit>, the KVS system downloads the user's voice profile into the KVS so that you can

train and recognize immediately after editing. When the phrase book is saved after training, all the pieces are uploaded from the KVS and the phrase book is written to the user's directory on the fixed disk under \KVSUSERS. (<edit> can also save the untrained KVS phrase book to disk.)

This means that you can interrupt that process of entering the language model to train the phrases entered so far. You can also interrupt training to do recognition testing. The <edit> command makes it easy for you to move from training to testing and back to editing. This ability to move back and forth is simpler with a language model in KVS format than with a language model in ASCII format. Features and functions of <edit> are discussed in Chapter 12.

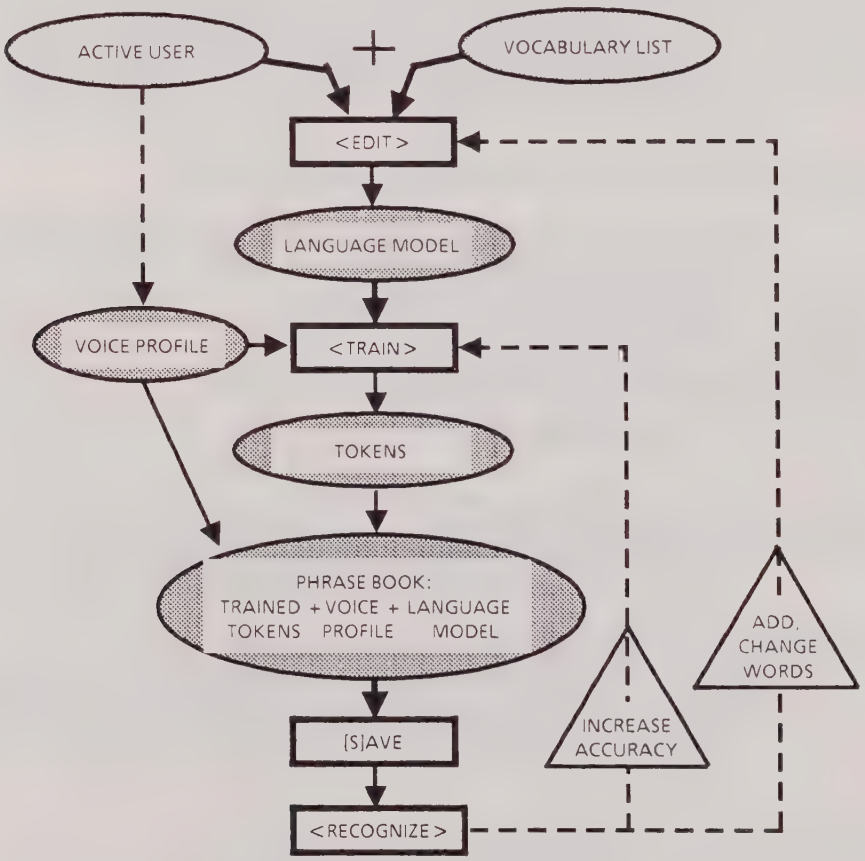

Fig. 6.2 - Phrasebook creation using <edit>

A SPREADSHEET PROGRAM

Using a spreadsheet program has three advantages over edit:

1. You can see more of your language model on the screen at once. (Since editing takes place in the 3-line KVS window, you can see at most 3 phrase-translation pairs on the screen at a time in <edit>.)
2. You can import vocabulary from an existing text file into the spreadsheet without retyping it.
3. You can see the contents of an unfamiliar language model you may want to change or adapt for another use.

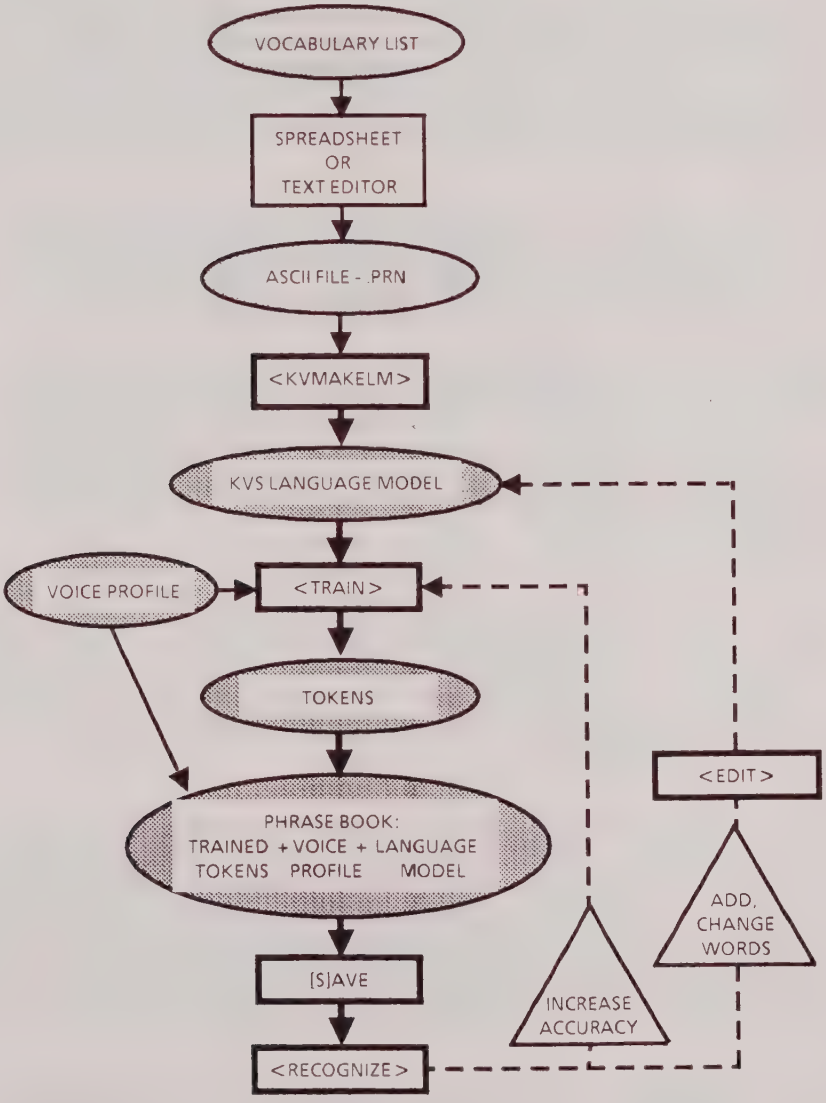

Fig. 6.3 - Phrasebook creation using external editor programs

Spreadsheet programs like 1-2-3 or its clones produce the ASCII format required by the KVS utilities, programs that run on the host PC and do not require a connected KVS. KVS utility programs, discussed in Chapter 9, are available to do the following:

- produce a language model in KVS format from one in ASCII (KVMAKELM);
- print a language model (KVLISTLM);
- produce an ASCII language model from one in KVS format (KVLISTLM).

The host language model utilities also have an option that can improve the quality of voice samples during training. This option randomizes the phrases the user is prompted to say during training so that the user will not be asked to say the same phrase many times in succession. A varied order assures more varied pronunciations in the set collected in training. The language model utilities are discussed in Chapter 9.

The phrase book creation cycle using a spreadsheet program on the host PC without a connected KVS is diagrammed in Figure 6.3. After producing a language model in ASCII format, you must run a utility (KVMAKELM) to compress it into KVS format.

The ASCII language model you build is saved in two forms: a .prn file (export format), which KVMAKELM accepts as input, and a .wks file, which the spreadsheet program accepts as input. The latter file makes it possible to go back into the spreadsheet to change the language model later, though the <edit> command is usually more convenient for making changes.

With the spreadsheet program, the phrase book is assembled in the KVS when a user gives the <train> command. <train> downloads the user's voice profile to the KVS so that tokens can be tried in recognition immediately after training. When the phrase book is saved, it is uploaded from the KVS and written to the user's directory under\KVSUSERS.

Not shown in Figure 6.3 is another way to make a phrase book. KVMAKEPB, a host utility, assembles a language model in KVS format with a specified voice profile and makes an untrained phrase book in a user's private directory. KVMAKEPB is run before a user trains. The resulting phrase book is missing only tokens. When the user trains the phrase book, training supplies the tokens. A developer may want to use KVMAKEPB to ready a phrase book for an end-user when

(a) the KVS is not connected and all the work of readying a phrase book for training is being done on the host PC;

(b) the language model is being prepared for several end-users. KVMAKEPB moves files into each end-user's directory without having to download and upload them between the PC and the KVS over the communications line. KVMAKEPB is faster than <edit> for this use.

A TEXT EDITOR

If you don't have a spreadsheet program and want to create a language model in ASCII format, you can imitate the output file format (export .prn) with any text editor. The .prn file (export format) is shown in Chapter 7.

The next chapters continue the discussion of language model making, training, testing, and changing.

Chapter 7 teaches you how to make a language model using each kind of tool.

Chapter 8 gives the rules for language models, ideas for effective language models, and examples.

Chapter 9 tells you how to run the KVS host utilities.

Chapter 10 discusses training, the last step before recognition. It also tells you how to add, delete, and replace tokens in a phrase book.

Chapter 11 tells you how to test your language model in recognition.

Chapter 12 documents <edit>. It tells you how to change a language model if recognition testing uncovers the need for change.

CHAPTER 7
TOOLS FOR MAKING LANGUAGE MODELS

CONTENTS:

- A Sample Language Model: Newdates
- Tutorial: How To Use <edit> To Make a Language Model
- Using WORDKIT with a Spreadsheet Program
- Tutorial: How to Use a Spreadsheet Program to Make a Language Model
- Tutorial: How to Use a Text Editor to Make a Language Model

A SAMPLE LANGUAGE MODEL: NEWDATES

This chapter shows you a sample language model and three ways to type it into the computer: with <edit>, with a spreadsheet program, and with a text editor. The language model you'll learn to build has 55 trainable phrases and lets you speak the date (month, day, year) to DOS. For example,

> You say: "set date"
> DOS types: Enter new date (mm-dd-yy):
> You say: "April two six eighty-six"
> DOS types: 4-26-1986 [RET]

Kurzweil AI supplies a few skeleton language models on the WORDKIT disk. This chapter demonstrates using two of the supplied language models -- WORDKIT and DATES -- to assemble the language model NEWDATES. The WORDKIT and DATES language models must be present on the fixed disk of your PC in the \KVSLMOD directory for you to use this tutorial. (Please see Chapter 2 for installation instructions.)

We'll begin by listing the words or phrases needed to set dates with DOS and to get a DOS directory. Next to each phrase, we'll list the translation DOS expects to receive.

PHRASE	TRANSLATION
"set date"	date\r
"directory"	dir\r
"directory wide"	dir /w\r

We will add these phrase-translation pairs to the NEWDATES language model after we assemble it. When assembled, NEWDATES will have vocabulary in several categories merged in from WORDKIT and DATES. The categories (subsets) are Command (voice commands), Digit, Yrslong (years of the decade), and Months (of the year).

We will add the DOS phrases to the Command subset and the Month subset so that a user can speak to DOS when the KVS is recognizing from either of these subsets.

Table 7.1 lists the assembled language model NEWDATES by subset. Each line contains one input (spoken phrase) paired with its output (translation string).

Table 7.1 SAMPLE LANGUAGE MODEL: NEWDATES

```
--generated from the recognition language model "wordkit"
--and compiled using kvmakelm Version 2.0;
--which contains 4 recognition subsets.
--There are 55 trainable phrases
--to be represented by 165 tokens (at resolution 4).
```

Phrases and translation strings in the "COMMAND" recognition subset:

1:	*UNKNOWN	<BEEP><WINDOW That word is not in the current vocabulary
2:	*ILLEGAL	<BEEP><WINDOW That word is not in the current subset>
3:	*ENTER	<WINDOW Listening for commands>
4:	#DELETE LAST WORD	\b
5:	#TRY AGAIN	\b
6:	#NEXT CHOICE	\b
7:	back1	\b
8:	microphone off	<microphone OFF><WINDOW Microphone is off>
9:	#LISTEN TO ME	<WINDOW Listening for commands>
10:	subset command	<subset command>
11:	subset digit	<subset digit>
12:	show vocabulary	<WINDOW Lstening for commands>
13:	where am I?	<subset>
14:	check microphone	<recognize .>
15:	train	<train .>
16:	pause	<pause>
17:	run KVTEST	kvtest\r
18:	exit	\032
19:	return	\r
20:	space	\s
21:	set date	date\r<subset months>
22:	directory	dir\r
23:	directory wide	dir /w\r

Phrases and translation strings in the "DIGIT" recognition subset:

1:	*UNKNOWN	<BEEP><WINDOW That word is not in the current vocabulary
2:	*ILLEGAL	<BEEP><WINDOW That word is not in the current subset>
3:	*ENTER	<WINDOW Listening for commands><parameter wordgap 3>
4:	*EXIT	<parameter wordgap 5>
5:	#DELETE LAST WORD	\b
6:	#TRY AGAIN	\b
7:	#NEXT CHOICE	\b
8:	back1	\b
9:	microphone off	<microphone OFF><WINDOW Microphone is off>
10:	#LISTEN TO ME	<WINDOW Listening for numbers>
11:	subset command	<subset command>
12:	subset digit	<subset digit>
13:	show vocabulary	<WINDOW Listening for numbers>
14:	where am I?	<subset>
15:	check microphone	<recognize .>

Table 7.1 (continued) SAMPLE LANGUAGE MODEL:NEWDATES

("DIGIT" subset, continued)

16:	train	<train .>
17:	pause	<pause.>
18:	run KVTEST	kvtest\r
19:	exit	\032
20:	return	\r
21:	zero	0
22:	oh	0
23:	one	1
24:	two	2
25:	three	3
.		.
.		.
.		.
31:	nine	9
32:	point	.
33:	space	\s
34:	set date	date\r<subset months>
35:	directory	dir\r
36:	directory wide	dir /w\r

Phrases and translation strings in the "YRSLONG" recognition subset:

1:	eighty	-1980\r
2:	eighty-one	-1981\r
3:	eighty-two	-1982\r
.		.
.		.
.		.
9:	eighty-eight	-1988\r
10:	eighty-nine	-1989\r
11:	ninety	-1990\r

Phrases and translation strings in the "MONTHS" recognition subset:

1:	*UNKNOWN	<BEEP><WINDOW That word is not in the current vocabulary>
2:	*ILLEGAL	<BEEP><WINDOW That word is not in the current subset>
3:	*ENTER	<WINDOW Listening for dates>
4:	#DELETE LAST WORD	\b
5:	#TRY AGAIN	\b
6:	#NEXT CHOICE	\b
7:	back1	\b
8:	microphone off	<microphone OFF><WINDOW Microphone is off>
9:	#LISTEN TO ME	<WINDOW Listening for dates>
10:	subset command	<subset command>
11:	subset digit	<subset digit>
12:	show vocabulary	<WINDOW Listening for dates>
13:	where am I?	<subset>
14:	check microphone	<recognize .>
15:	train	<train .>

Table 7.1 (continued) SAMPLE LANGUAGE MODEL: NEWDATES

("MONTHS" subset, continued)

16:	pause	<pause>
17:	run KVTEST	kvtest\r
18:	exit	\032
19:	return	\r
20:	zero	0
21:	oh	0
22:	one	1
23:	two	2
24:	three	3
.		.
.		.
.		.
30:	nine	9
31:	point	.
32:	space	\s
33:	eighty	-1980\r
34:	eighty-one	-1981\r
35:	eighty-two	-1982\r
.		.
.		.
.		.
41:	eighty-eight	-1988\r
42:	eighty-nine	-1989\r
43:	ninety	-1990\r
44:	January	1-
45:	Feburary	2-
46:	March	3-
47:	April	4-
.		.
.		.
.		.
54:	November	11-
55:	December	12-
56:	set date	date\r<subset months>
57:	directory	dir\r
58:	directory wide	dir /w\r

* The entries beginning with * and # are described in Chapter 8 under "Star Phrases and Voice Commands."

Notice that a phrase can appear in more than one subset. This means it has more than one translation. How does the KVS know which translation to use?

When you start recognition, the KVS translates from the first recognition subset by default (subset Command in NEWDATES). After that, the <subset {name}> command tells the KVS which subset to translate from.

That command can be typed at the keyboard or written into the language model to be spoken. If it is written into the language model, a user can switch subsets by voice through the <subset> command embedded in a translation string. If we expand the translation of "set date" to include a <subset> command such as

date\r<subset months>

when the user says "set date," DOS will get the date command and the KVS will switch to recognizing from the Months subset (see Table 7.1). Chapter 8 gives more information about phrases, translations, subsets, and other elements of a language model.

TUTORIAL: HOW TO USE <edit> TO MAKE A LANGUAGE MODEL

<edit> is the KVS on-board phrase book editing program. It is convenient for creating a language model from scratch, assembling a language model from existing language models, or modifying an already created language model.

The following steps are required to assemble a language model using <edit>. Each step is described in detail in this section using the NEWDATES language model.

1. With paper and pencil, write out the language model you want to create (optional).

2. Sign on to the KVS and invoke the <edit> command with the name of a new (empty) phrase book (NEWDATES).

3. Merge WORDKIT's subsets into NEWDATES by choosing [F] (the files menu) from the main <edit> menu and then selecting [M] (merge phrase book). Then merge DATES' subsets into NEWDATES in the same way.

4. Augment the vocabulary in the Months subset by choosing [C] (copy subset) from the main <edit> menu. Into subset Months copy translations for the years of the decade from subset Yrslong, for the digits from subset Digit, and for voice commands from subset Command.

5. Edit some of the voice command translations to make them appropriate for subset Months. Choose [E] (edit phrase) from the <edit> menu, and correct the translation in the Months subset.

6. Add the DOS vocabulary to the language model by choosing [A] (add phrase) from the main <edit> menu and filling in the phrase names and translations for each subset.

7. Optionally, save the language model by choosing [F] (files menu) from the main <edit> menu and then selecting [S] (save phrase book).

8. Train the language model by choosing [RET] (train) from the main <edit> menu and [S] (save tokens)to disk at the end.

When the phrase book is trained, test the language model by pressing [RET] for recognition after training. Try setting the date, turning the microphone on and off, running KVTEST, and speaking the number words. You can invoke <edit> again to change the language model after testing or invoke <train> to add tokens of any confusable phrases.

Step 1: Write Out the Phrases and Translations in the Vocabulary

With paper and pencil, write out the phrase-translation pairs you want in your language model as we did in writing out the DOS commands (optional).

Step 2: Sign On to the KVS and Invoke the <edit> Command

To use <edit>, you need a connected KVS. Download the KVS software, sign on, and invoke <edit> by typing

```
KVLOAD [RET]
<user {your name}>
<edit NEWDATES>
```

<edit>, as a KVS control command, communicates with you through the KVS three-line window at the bottom of the screen. Since the phrase book doesn't exist, <edit> replies, for example:

```
There is no phrase book newdates
Press [Y] to create one
      [N] to exit
```

Press [Y] for yes. Edit now creates an empty phrase book containing only your ENROLL voice profile, which it downloads, displaying these messages:

```
Creating Phrase Book Newdates
Please stand by . . .
Loading Voice Profile ENROLL
Please stand by. . .
```

When the preparation of your new phrase book is complete, page 1 of the <edit> menu is displayed in the window:

EDIT MENU: page 1 of 2	**Phrase Book: Dates**	**[N] Next menu page mic OFF**
[A] Add phrases	**[E] Edit phrases**	**[V] Vocabulary list menu**
[D] Delete phrases	**[F]Files**	**[RET] Training**

Notice that there are 2 pages to the edit menu. Look at the next menu page by choosing [N]:

EDIT MENU; page 2 of 2	Phrase Book: Dates	[N] Next menu page mic Off
[R] Remove subsets	[S] Setup menu	[X] Change subset spelling
[D] Delete phrases	[I] Information	[C] Copy subset

Step 3: Merge WORDKIT's Vocabulary into Your Phrase Book

Select [F] (the files menu) from the main menu. After you press [F], you'll see a new menu.

FILES MENU	Phrase Book: Newdates	
[L] List phrase books	[D] Delete phrase book	[M] Merge phrase book
[S] Save phrase book	[P] Publish language model	[RET] for main menu

Press [M] (merge phrase book). Name WORDKIT when Merge asks for a phrase book name. The WORDKIT language model will then be read from disk and included in NEWDATES. WORDKIT is the "source" for the merge operation. NEWDATES is the "destination." Here's how to do it: To the prompt "Type phrase book name:" type

WORDKIT [RET]

WORDKIT is read into the KVS now, but no merge has taken place yet. When the KVS asks "Which subsets?" to merge, enter [^W] (the wild card) and [RET] to include both of WORDKIT's subsets, Command and Digit.

Before performing the operation, the KVS reports the number of subsets found in WORDKIT. When the KVS prompts "Press [RET] to accept the match or [ESC] to try again," press [RET]. After that, the merge operation is carried out. Press [RET] again to get back to the "Type phrase book name" prompt so you can do another merge.

This time, the DATES language model is the "source" from which to merge subsets. Type DATES [RET] and another [^W] (wild card) to merge both of DATES' subsets, Yrslong and Months. Press [CTRL-Z] to return to the main <edit> menu.

The NEWDATES language model now has a vocabulary of 52 trainable phrases and three star phrases, but some phrase-translation pairs are incomplete. Below is an excerpt showing the subsets of the language model side by side so you can compare what they contain at this stage. Notice that subset Months has translations only for the months of the year. This means that the KVS won't recognize you if you say, for example, "show vocabulary" when Months is the current subset. Our next step will be to make the digit and voice command vocabulary available in the Months recognition subset.

EXCERPT FROM NEWDATES LANGUAGE MODEL (BEFORE COPY SUBSET)

	PHRASE	Command	Digit	Yrslong	Months
	.	.	.	.	.
	.	.	.	.	.
9:	#LISTEN TO ME	<WINDOW...	<WINDOW...		
10:	subset command	<subset command>	<subset command>		
11:	subset digit	<subset digit>	<subset digit>		
12:	show vocabulary	<WINDOW ...	<WINDOW ...	.	
	.	.	.	.	.
	.	.	.	.	.
20:	zero		0		
21:	oh		0		
22:	one		1		
23:	two		2		
24:	three		3		
	.	.	.	.	.
	.	.	.	.	.
30:	nine		9		
31:	point	.	.		
32:	space	\s	\s		
33:	eighty			-1980\r	
34:	eighty-one			-1981\r	
35:	eighty-two			-1982\r	
	.	.	.	.	.
	.	.	.	.	.
41:	eighty-eight			-1988\r	
42:	eighty-nine			-1989\r	
43:	ninety			-1990\r	
44:	January				1-
45:	Feburary				2-
46:	March				3-

Step 4: Copy Subsets

We will copy translations of voice commands, digits, and years of the decade to the Months recognition subset so that this vocabulary will be available to us when we are setting a date.

Select [C] (copy subset) from the main <edit> menu. When the KVS asks which "Subset to copy from" (the source), enter Command. When the KVS asks which "Subset to copy into," enter Months (the destination). Repeat the [C] (copy subset) operation three times. Here's what to type each time you are prompted for a source and a destination.

Command [RET]
Months [RET]

Digit [RET]
Months [RET]

Yrslong [RET]
Months [RET]

Below is an excerpt from NEWDATES showing what it contains after the copy is performed. Voice commands and numbers will now be recognized if spoken when the KVS is recognizing from subset Months.

EXCERPT FROM NEWDATES LANGUAGE MODEL (AFTER COPY SUBSET)

	PHRASE	Command	Digit	Yrslong	Months
	.	.	.	.	.
	.	.	.	.	.
9:	#LISTEN TO ME	<WINDOW...	<WINDOW...		<WINDOW...
10:	subset command	<subset comm..	<subset comm..		<subset comm..
11:	subset digit	<subset digit>	<subset digit>		<subset digit>
12:	show vocabulary	<WINDOW...	<WINDOW...		<WINDOW...
	.	.	.	.	.
	.	.	.	.	.
20:	zero		0		0
21:	oh		0		0
22:	one		1		1
23:	two		2		2
24:	three		3		3
	.	.	.	.	.
	.	.	.	.	.
30:	nine		9		9
31:	point	.	.		.
32:	space	\s	\s		\s
33:	eighty			-1980\r	-1980\r
34:	eighty-one			-1981\r	1981\r
35:	eighty-two			-1982\r	1982\r
	.	.	.	.	.
	.	.	.	.	.
41:	eighty-eight			-1988\r	-1988\r
42:	eighty-nine			-1989\r	-1989\r
43:	ninety			-1990\r	-1990\r
44:	January				1-
45:	February				2-
46:	March				3-

Step 5: Edit Translations

When an already entered field needs correcting, use [E] (edit phrase). We need to edit the translations for *ENTER, #LISTEN TO ME, and "show vocabulary" in the Months subset since the window messages they post are incorrect for Months. Here's how to edit the translations:

1. Press [CTRL-Z] to return to the main <edit> menu.
2. Select [E] (edit phrases).
3. Type *ENTER [RET] when [E] asks "which phrase?"
4. Press [RET] again when [E] echoes the phrase.
5. Press [CTRL-S] to skip to Months, the fourth recognition subset.
6. Type Months [RET] when [E] asks for a subset name.
7. Entry of [RET] gives a new window that prompts for changes to the translation of the phrase in subset Months.
8. Since we don't want a window message from the KVS when a user says "set date," we will clear the translation field. Press [CTRL-C] and [RET] to delete the <window> command.
9. Press [RET] again to return to the beginning of [E]. Type #LISTEN TO ME, the next phrase to correct, and [RET]. Repeat steps 4-7 above.
10. Edit the translation of #LISTEN TO ME in the Months subset so that it reads <WINDOW Listening for dates>. To do this, delete the word "commands" with the [DEL] key and type in the word "dates." Press [RET] to enter the corrected translation.
11. Press [RET] again to return to the beginning of [E]. Type "show vocabulary," the next phrase to correct, and [RET]. Repeat steps 4-7 above.
12. Edit the translation string for "show vocabulary" in the Months subset so that it reads <WINDOW Listening for Dates>, as you did in step 10 above. This is the final correction.
13. Press [CTRL-Z] to return to the main <edit> menu.

Step 6: Add Phrases and Translations to the Language Model

Select [A] (add phrases) from the main <edit> menu to start the addition of DOS commands. We will add the phrase-translation pairs "set date," "directory," and "directory wide" as shown in Table 7.1. An add phrase operation consists of adding a phrase and its translation in every subset where the phrase will be recognized.

After you press [A], you'll see this screen:

```
ADD; [RET] to Enter          [ESC] to exit          [^0] for options
Type PHRASE 1:
```

For each line you must type

1. the phrase to add;

2. its translation in the Command subset;

3. its translation in the Digit subset;

4, its translation in the Months subset.

Type the first phrase "set date," followed by [RET] to enter it into the KVS. Entry of [RET] gives the next window, which prompts for the translation string:

```
ADD: [TAB] same as phrase   [RET] to enter   [^0] for options   [^S] choose subset
PHRASE 1: *set date         Subset 1: Command
Type translation:
```

Type (from Table 7.1)

date\r<subset months> [RET]

This is the translation of "set date" in the Command subset. Entry of [RET] gives a prompt to enter a translation for "set date" in the Digit subset. Type the same translation:

date\r<subset months> [RET]

Entry of [RET] gives a prompt to enter the translation for "set dates" in the Months subset. This time type

date\r [RET]

Type [RET] twice to specify another phrase to add.

The next phrase we want to add is "directory." Refer to Table 7.1 and type in the phrase and its translation in each subset. Do the same for the phrase "directory wide." You can continue to add phrases to the language

model until you press [CTRL-Z], which returns you to the main <edit> menu.

Shortcuts

Several shortcuts to filling in translation fields are mentioned here. They are also listed in full in Table 12.3, and in the <edit> command summary in Appendix B.

TAB

A reply of [TAB] to "Type translation" inserts the string you typed for PHRASE in the translation field. You can then press [RET] to enter it as is or you can change it.

CTRL-P (put), CTRL-G (get)

If the translation is a long string and is used again in another subset, <edit> can remember it and retrieve it from memory on request. To use the remember function, first type the translation but don't press [RET]. Then type CTRL-P to put (store) the translation in memory. Press [RET] to go to the next subset. Type CTRL-G to get the translation from storage and insert it in the translation field in the current subset. This is what you type:

(translation string)
CTRL-P [RET]
CTRL-G [RET]

CTRL-G gets the translation from storage and displays it in the window. [RET] enters it in the language model.

CORRECTING TYPING MISTAKES: CTRL-C FOR CLEAR

If you want to correct a mistake in the field you're keying in, use [backspace] to erase the last character or CTRL-C to clear the field. Additional commands for deleting and moving the cursor in a field are listed in Table 12-3 and displayed on the screen when you press CTRL-O (options).

Wild Cards and String Completions

In the [E] (edit) function, a wild card is provided to help you ask for fields whose names or contents you can't remember or multiple fields with a common part to their names. The wild card character is CTRL-W [^W]. If you type [^W] when [E] asks "Which phrase?" all phrases in the phrase book will be presented for editing, starting with phrase number 1. If you know some but not all parts of a phrase you want to edit, you can use a wild card for any parts you don't know. Table 7.2 shows how to use wild cards to search for phrases you want to edit. All searches are case-insensitive.

Alternatively, if you know the beginning of a phrase, subset name, or a translation, type as much as you know and then type a question mark (?). <edit> will fill in the rest.

Table 7.2 -- USE OF WILD CARDS TO SEARCH A LANGUAGE MODEL

Where to put [^W] in a search string	Examples of phrases found by <edit>
[^W]	All phrases (or subsets, or translations)
Mode[^W]	Modest Mode change
[^W]mode	Command mode Date Mode
[^W]mode[^W]	KVS Model One
[^W]ee[^W]	Milwaukee Queens

Step 7: Save the Language Model (Optional)

Once the language model is keyed in to a phrase book, you may want to save the phrase book to disk:

1. Press CTRL-Z to return to the main <edit> menu.
2. Press [F] to see the files menu.
3. Choose [S] to save the phrase book to disk.

4. Press [RET] to accept NEWDATES as the name for the phrase book you are saving.

The [S] (save phrase book) function creates an untrained phrase book in your private directory under \KVSUSERS. When the phrase book has been saved, the files menu returns. You can save the phrase book periodically as you go along to avoid losing work due to power failures, loose cable connections, or any other unpredictable event.

Step 8: Train the Language Model

Your last step before using the phrase book for recognition is to train each of the vocabulary phrases. Training creates a file of tokens to complete the phrase book. To start the training process, with the main <edit> menu on the screen, press [RET]. The <train> menu appears. When you finish training, you'll save your tokens. Then you can come back to <edit> if you need to make changes, or you can go on to recognition. The discussion of training continues in Chapter 10.

USING WORDKIT WITH A SPREADSHEET PROGRAM

Using a spreadsheet program such as Lotus 1-2-3 is another easy way to design a language model because phrase-translation pairs can be seen side by side in columns.

The spreadsheet file, wordkit.wks, on your WORDKIT disk can be included in a program such as 1-2-3 with the /[f]ile [r]etrieve operation and then tailored to your application. Or you can protect the original of wordkit.wks by copying it over to a new name and retrieving [r] the copy to work with.

WORDKIT also contains some lists of common words already typed. If your application has a use for any of the pretyped words, you can import them into the spreadsheet with a /[f]ile [i]mport command. The pretyped vocabulary includes the digits as English words, the digits as numerals, the phonetic alphabet, the alphabet as letters, names of the months, and the names and abbreviations of the states. A directory of **\123files** on your WORDKIT disk will show you the names of these files. Each ends in the suffix .prn, which is the spreadsheet program's designation for files in ASCII (printable) format. .prn files can be included in Column A, the PHRASE column, or in a subset column. That is, they can be used as spoken phrases or translations.

To create a language model with WORDKIT, after it is installed, you simply copy wordkit.wks to a new file and then use a spreadsheet program like 1-2-3 to add application-specific phrases and translations.

In the spreadsheet program, a single letter is usually sufficient to specify a command. The tutorial that follows uses 1-2-3 as an example and shows the sufficient letter in square brackets. For example, the notation

[in 123] /[f]ile [r]etrieve newdates [RET]

means that to retrieve a file in 1-2-3 or its clones, you type

/ f r

and the name of the file -- in this case, newdates -- followed by a carriage return.

TUTORIAL: HOW TO USE A SPREADSHEET PROGRAM TO MAKE A LANGUAGE MODEL

We will use wordkit.wks to create the language model NEWDATES and then turn it into a phrase book that will allow us to set the date (month, day, year) with DOS (see Table 7.1).

The steps in the procedure are previewed here and then explained separately.

1. Change the KVS command delimiter from open angle bracket (<) to something the spreadsheet program doesn't use, like at sign (@).

2. Load wordkit.wks into the spreadsheet.

3. Add years ('80 - '90) to PHRASE column (A).

4. Add names of the months to PHRASE column (A).

5. Verify that days of the month (number words) are already typed in the PHRASE column (cells A21..A31) (optional).

6. Add three DOS commands to the PHRASE column.

7. Make Column E a recognition subset labeled Dates.

8. Enter translations (arabic numbers) for the days of the month at cells E21..E31.

9. Add translations for the years at cells E35..E45.

8. Add translations for the months at cells E46..E57.

10 Add translations for voice commands at cells E2..E20 and DOS commands at cells E58..E60.

11. Save the new language model as a print file, and save the worksheet.

12. Restore the KVS command delimiter to open angle bracket (<).

Step 1: Change the KVS Command Delimiter

Before starting this procedure, type

<PARAM BRACKETS @>

This allows you to type the angle bracket character < in a spreadsheet cell without causing the KVS information window to pop up.

Step 2: Load WORDKIT into the Spreadsheet Program

First make a copy of wordkit.wks and enter the spreadsheet program:

C\> copy \123files\wordkit.wks c:\123\newdates.wks
C\> 123
[in 123] /[f]ile [r]etrieve newdates [RET]
[123] /[w]orksheet [g]lobal, [c]olumn width [RET] 60 [RET]

The /[f]ile [r]etrieve command imports WORDKIT, renamed as NEWDATES, into your blank spreadsheet. As you can see, you already have a vocabulary that includes voice control words (cells A2..20) and the digits (A21..A31).

Your task is to add the names of the 12 months and create translations that type hyphens around the day of the month without the user having to say "hyphen." The language model can automatically affix hyphens to the translations for the months and years if the phrase "April" translates to "4-" and the phrase "86" translates to "-1986\r" (see steps 3, 4).

Steps 3, 4: Add the Years of the Decade and the Names of the Months

Next we would like to add the years of the decade to the PHRASE column (A). The first free cell is at A35. We will include the text file yrslong.prn starting at A35. We would also like to include the names of the months (months.prn) to the PHRASE column. We will include the text file, months.prn, 10 cells lower at A46.

[in 123] [F5] A35 [RET]
[in 123] /[f]ile [i]mport [t]ext yrslong [RET]
[in 123] [F5] A46 [RET]
[in 123] /[f]ile [i]mport [t]ext months [RET]

Step 5: Verify Days of the Month (Number Words) in the PHRASE Column

WORDKIT already contains the days of the month as number words (cells A21..A31); you don't have to type them. To verify this, press [F5] (go to) and then type A21. The other way to move the cursor to a specific cell is with the arrow keys.

Step 6: Add Three DOS Commands to the PHRASE Column

We will add DOS commands to set the date by voice, get a directory, and get a directory in wide format.

We can include a voice command to set the date by entering the phrase "set date" in cell A58 of the PHRASE column and entering a translation in cell C58 of the Command subset that will switch into the Dates recognition subset when this phrase is spoken.

We can include voice commands to get a DOS directory by entering the phrases "directory" and "directory wide" in cells A59 and A60 and entering their translations in the Command subset:

```
[in 123]   [F5] A58 [RET]
[in 123]   set date [RET]
[in 123]   [F5] C58 [RET]
[in 123]   date\r <subset dates> [RET]
[in 123]   [F5] A59 [RET]
[in 123]   directory [RET]
[in 123]   [F5] C59 [RET]
[in 123]   dir\r [RET]
[in 123]   [F5] A60 [RET]
[in 123]   directory wide [RET]
[in 123]   [F5] C60 [RET]
[in 123]   dir /w\r [RET]
```

Step 7: Make Column E a Recognition Subset Labeled Dates

Next we want to create a subset called Dates in which we will add translations for the appropriate phrases: voice commands, the years, the months, and the digits 0-9. We create the subset by typing Dates in E1, the first free column.

```
[in 123]   [F5] E1 [RET]
[in 123]   Dates [RET]
```

Step 8 Enter Translations (Arabic Numbers) for the Days of the Month at Cells E21..E31

Next we fill in the translations for the digits. Number words for the digits are located at cells A21 .. A31, so we want to add our translations at E21 .. E31. Since arabic numerals have already been typed into the worksheet at D21 .. D31, we can use 1-2-3's /[c]opy command to copy these translations to column E. Be sure you have typed the right range and destination before giving the /[c]opy command.

```
[in 123]   [F5] D21 [RET]
[in 123]   /[c]opy D21..D31 [RET] E21 [RET]
```

Step 9: Add Translations for the Years at E35

Next we want to enter the translations for the years. The translation for "80" will be "-1980\r." Since all the translations will look similar to this, we can save some typing by copying "-198" to all cells and then editing the cells to add the "0," "1," "2," etc. and \r (carriage return). We type a single quote before the "-198" to let 1-2-3 know that we are entering text, not data. We use the [F2] key to edit the text in the active cell.

```
[in 123]    [F5] E35 [RET]
[in 123]    '-198 [RET]
[in 123]    /[c]opy E35 [RET] E36..E45 [RET]
[in 123]    [F2] 0\r [↓]
[in 123]    [F2] 1\r [↓]
        etc. through cell E45
[in 123]    [F2] [DEL] 90\r [RET]
```

Step 10: Add Translations for the Months at E46

Next we want to add translations for the months. Let's move to E46 and add the translation for January, '1-.

```
[in 123]    [F5] E46 [RET]
[in 123]    '1- [↓]
[in 123]    '2- [↓]
        etc., through cell E57
[in 123]    '12- [↓]
```

Step 11: Add Translations of Voice Commands and DOS Commands

We will copy translations of WORDKIT's voice commands and then edit the translation for *ENTER (A6), #LISTEN TO ME (A14), and show vocabulary (A18). We will also add translations of the three DOS commands.

```
[in 123]    /[c]opy C2..C21 [RET] E2 [RET]
[in 123]    [F5] E4 [RET]
[in 123]    '<WINDOW Listening for dates>[RET]
[in 123]    [F5] E10 [RET]
[in 123]    '<WINDOW Listening for dates>[RET]
[in 123]    [F5] E13 [RET]
[in 123]    '<WINDOW Listening for dates>[RET]
[in 123]    [F5] E58 [RET]
[in 123]    date\r<subset dates>[RET]
[in 123]    [F5] E59 [RET]
[in 123]    dir\r[RET]
[in 123]    [F5] E60 [RET]
[in 123]    dir /w\r[RET]
```

Step 12: Save the New Language Model as a Print File and Save the Worksheet

Make a .prn file from your spreadsheet as a cell-formula, unformatted file:

```
[in 123]   [p]rint [f]ile dates [RET]
[in 123]   [r]ange A1..E60 [RET]
[in 123]   [o]ptions [o]ther [c]ell formulas [o]ther
           [u]nformatted
[in 123]   /[q]uit [g]o [q]uit
[in 123]   [f]iles [s]ave [RET]
[in 123]   /[q]uit [y]es
```

Step 13: Restore the KVS Command Delimiter

Restore the KVS command delimiter to open angle bracket by typing

@PARAM BRACKETS < >

Now you have a list of phrases and translations in the ASCII form needed for KVMAKELM, the utility that will transform it into a language model. You have saved your worksheet as two files: NEWDATES.PRN (to be input to KVMAKELM), and NEWDATES.WKS, the input to the spreadsheet program if you ever want to change the language model with the spreadsheet program instead of with <edit>.

The next step in the procedure is to make a language model in KVS format. Chapter 9 tells you how to do this. If you skip to Chapter 9, be sure to come back and study the material in Chapter 8. Chapter 8 has basic information about how to make any language model more powerful and easier to use.

TUTORIAL: HOW TO USE A TEXT EDITOR TO MAKE A LANGUAGE MODEL

If you are not using <edit> and don't have a spreadsheet program, you can make a language model file for input to KVMAKELM by imitating the spreadsheet program's .prn file (export) format.

This section describes how to create that format with a text editor. Figure 7.1 shows an excerpt of the export .prn file format for the sample language model NEWDATES.

```
A1: 'PHRASE
B1: 'Command
C1: 'Digit
D1: 'Dates
A2: '*UNKNOWN
B2: '<beep><window That word is not in the current vocabulary>
C2: '<beep><window That word is not in the current vocabulary>
D2:
A3: '*ILLEGAL
B3: '<beep><window That word is not in the current subset>
C3: '<beep><window That word is not in the current subset>
D3: '<beep><window That word is not in the current subset>
```

Figure 7.1 Export .prn File Format of Dates Language Model

The following are the rules for laying out a language model with paper and pencil and typing it into a text editor.

1. Label each column with a letter from left to right. Begin with A. After Z, begin with AA, AB, AC, AD...BA, BB, BC, BD and so on. Label each row from top to bottom with a number. Each entry in the table -- a cell -- then has an address label, which is a letter followed by a number.

2. Use the top cell in each column (cells A1, B1, etc.) for the name of the column (subset).

3. The first column is always the column of inputs and must have the name PHRASE. Output columns must be to the right and can have any name you choose.

4. All non-alphanumeric characters (carriage return, tab, etc.) should be entered in the octal backslash format. A complete list appears in Appendix A. Function keys and escape and control sequences should be entered as hexadecimal codes. The list of function key codes also appears in Appendix A.

5. Type one cell from the language model per line in the text file.

6. Begin each line with the address label of the cell.

7. Enter the cells in row-major order: that is, all the columns in one row, then all the columns in the next row, and so on. For example A1, B1, C1, D1, A2, B2, C2, D2, etc.

8. Type these characters after the address label: a colon, a space, an apostrophe (optional), and the contents of the cell.

9. Don't skip any column letters in row 1.

10. Don't skip any row numbers in entering successive rows.

CHAPTER 8
LANGUAGE MODEL DESIGN

CONTENTS:

- What Goes into a Language Model?
- How Big Can a Language Model be?
- Considerations for Language Model Design
- Examples

WHAT GOES INTO A LANGUAGE MODEL?

So far, we have been talking about language models as if they contained just those phrases and translations that control an application program or enter data to it.

You can write other elements into a language model to make it more effective and easier to use. This chapter introduces those elements, discusses requirements of good language models, and gives examples of how to write voice commands into a language model for end-users' convenience.

The elements that can be included in a language model are as follows:

- spoken phrases and their character string translations;
- names of KVS software routines and their arguments (star and sharp phrases);
- translation subsets;
- sharp subsets (containing spoken responses to the <edit> command and other KVS angle bracket commands).

Choosing Spoken Phrases

A vocabulary developer designs a general-purpose base vocabulary to serve as a starting point for all users of the application. The base vocabulary contains the basic phrases to do the following:

- control application software, such as order entry or inventory management, by voice (see Table 8.3);
- control the KVS (see Table 8.1);
- correct errors in recognition or mistakes in speaking (see Table 8.2);
- display help messages (see Table 8.4).

In many cases, the base vocabulary uses no more than 50% of the 1,000-word allotment for a language model. Once the base vocabulary is designed, it can be expanded and tailored for a user site by adding site-specific phrases. Such phrases may be

- part numbers or part descriptions used for data entry at the site,
- phrases to control the computing environment at the site, or

- phrases to invoke and use utility programs bound to PC function keys, such as voice mail.

The same basic vocabulary can be distributed to any user site that runs the application.

A site-customized vocabulary can be further modified by end-users for their own convenience. The <edit> command, discussed in Chapter 12, lets an end-user add words to an existing vocabulary at any time.

When choosing vocabulary, it's important to know how to run the application and to have input from the end-user. Chosen phrases should be familiar, easily understood, and descriptive of their function so users can understand and remember them. The phrase "Lotus edit" would be a better choice than "F1." Or "rear tulip panel" is a better choice than its part number, "A245."

A single translation can be invoked by many different words (synonyms). For example, a language model can include four phrases to type a carriage return: "return," "process," "enter," or "OK." Providing synonyms brings a new flexibility to the strict rules for input that most computer programs depend on. Use of synonyms lets different speakers choose the phrase they feel most comfortable with.

Through the use of multiple subsets in a language model, the same phrase can return different translations in different contexts (see "Subsets," below).

Constraints on vocabulary selection come from the demands of the application, the user, and the recognizer itself. No phrase can take longer to say than 2.0 seconds. Very short phrases may occasionally be missed. These considerations and others are discussed under " Considerations for Language Model Design," below.

Valid Phrases

Phrases are the words in the PHRASE column (A) of a language model -- those things that the user will say. They must conform to the following rules:

1. Phrases may be made up of any printable characters from the ASCII set. Internal spaces are allowed.

2. Phrases may be up to 45 characters long.

3. A phrase must not begin with a star (*) or a sharp (#) unless it is one of the KVS reserved names or labels (see Tables 8.1 and 8.2).

KVS software is not case-sensitive to phrase names or subset names. When you are creating or changing a language model and specifying a phrase name, <edit> ignores differences in case.

Defining Translation Strings

A translation string for a phrase may be a message for the host computer, a message for the KVS, or both. A message may be data,or a command.

KVS angle bracket commands are run-time messages sent to the KVS. Delimited by the special characters < and >, they control KVS operation and are used, for example, to sign a user on and off, to initiate recognition and training, to obtain help and status reports, and to change parameters controlling recognition. Angle bracket commands are summarized in Appendix B.

A sample translation with messages for the host computer and an angle bracket command for the KVS is shown below. The translation of the phrase "new sentence" tells the host to type a backspace, a period and two spaces and tells the KVS to switch into a new subset where the first letter of each word is capitalized:

PHRASE	TRANSLATION
new sentence	\bs .\s\s<subset caps>

When a translation contains more than one message, the KVS sends them one at a time in sequence from left to right. When the messages are multiple commands, they are executed in order from left to right, and, as in a batch procedure, a subsequent command is not begun until the previous one finishes.

KVS angle bracket commands beginning with ! (exclamation mark) are an exception. An ! inside an angle bracket command sends the command immediately, without waiting for the previous one to complete. For example, the translation of the phrase "add phrase" calls <edit> and simultaneously passes to <edit> the character [a] to select [A] (add phrase) from the <edit> menu. Without the !, the <choose> command would not be executed until the user had exited from <edit> into training or recognition.

PHRASE	TRANSLATION
add phrase	<edit .><!choose a>

The only limit to the number of messages that can be concatenated in one translation string is the number that will fit in the 498-character limit for translation strings.

Valid Translation Strings

The following are allowed in translation strings:

1. Printable ASCII characters.

2. Nonprinting characters such as space or carriage return, entered in backslash format and octal notation (\s, \r or \040, \015). Leading zeros may be omitted from octal strings: \012 and \12 are both valid and mean the same thing. (Include the leading zero to avoid

ambiguity when the character following the octal code in the translation string is a digit.)

The backslash is the only character that can prefix an octal code,or the C abbreviation for such nonprinting characters as \r [RET]. (Table A-3 in Appendix A lists the C abbreviations for nonprinting keys.) Use backquote before all letters except the C abbreviations.

NOTE

Letter abbreviations for the nonprinting characters "r," "e," "n," and "t" must be lowercase; for example \r; not \R.

The quote character by default is a single backquote character (`). The use of backquote as the quote character is changeable with <param quotechar>.

What must be quoted?

(a) All octal codes (see Appendix A).

(b) Any character that would normally terminate an angle bracket command must be quoted if it is part of a <choose> string. This includes [RET], [ESC], and close angle bracket, >. (Angle brackets are the default delimiters but can be changed with <param brackets>.)

(c) The backquote character if you want to send a backquote to the host.

(d) The backslash character itself when

(i) you want it to show up in the KVS window, or

(ii) you want to send a backslash to the host.

To quote the backslash character, double it.

For example, if you are entering a language model with the <edit> command and type this translation string:

<!choose aroot \\r cd\\ `ctrl-z><edit>

the backslash is doubled for \r, [RET] because \r must show up in the <edit> window when you display translations. Backslash is doubled following cd\\ because you want to pass cd\ to the host. [ctrl-z] is quoted so it doesn't switch you to the main<edit>menu when you enter it as a translation string. Do not double the backslash for \r, [RET] in a choose string sent with KVSEND

4. Extended ASCII representation of the function keys and keypad plus key sequences beginning with SHIFT, CTRL, and ALT are all entered in C backslash format and octal notation (F1 is \073). The tables in Appendix A list the octal equivalents of IBM PC

alphanumeric keys, function key codes, keypad, and key sequences. Use the tables when you use a spreadsheet program to create a language model. To add translations with <edit>, just press backquote (`) followed by the key and the right code will appear.

5. If you use a spreadsheet program to type in your language model, you can add final spaces to translation strings only by typing \s once for each space. For example, to enter the phrase or translation "name_._._.", you would type "name\s\s\s." Final spaces typed in with the space bar are ignored by 1-2-3 and its clones.

6. Two characters are reserved for KVS control command delimiters: < and >. To use a command delimiter in its literal sense, quote it. For example, to use > as the MS-DOS redirection command switch, precede it with the backquote character. Backquote (`) is the KVS default quote character.

NOTE

Type translations exactly as required by the application when the application is case-sensitive.

Star and Sharp Phrases

Phrases that begin with * or #, such as *ENTER or #DELETE LAST WORD, are really names of KVS software routines. Star phrases are never spoken by users; they simply invoke routines. Sharp phrases also invoke routines, but they are used in another way as well. Sharp phrases show up in training as prompts, and a user must respond to them. The user may say the sharp phrase or choose another word to say instead. For example, when prompted to say #DELETE LAST WORD, a user may say "delete last word," "scratch that," or any desired phrase. If the user says "scratch that," then "scratch that" becomes the command to delete the last word from the screen. The chosen phrase must be trained and used consistently.

Star routines are invoked automatically when the KVS is in a certain processing state. For example, the routine named *ENTER is invoked when the KVS switches to a new translation subset.

Sharp routines are invoked when the KVS recognizes the word associated in training with the sharp phrase. For example, the routine #DELETE LAST WORD is invoked when the KVS recognizes "scratch that."

The translation string you enter in the language model for a star phrase or a sharp phrase is sent to the KVS. The KVS passes it as an argument to the software routine.

Tables 8.1 and 8.2 list all star and sharp routines by name and for each routine give the function, how the routine is invoked, and a sample argument (translation string entry). To use the tables, look up the

functions you want and then look in the reference section that follows the table. The reference section devotes a page to each star or sharp phrase. The reference pages explain how the routines work, when to use them and how to write them into a language model. WORDKIT (see Chapter 7) uses many of these phrases.

Subsets

In designing a language model, you define different tasks that a user of an application program does and different contexts that the program offers for such tasks. The divisions you define can correspond to menus in a menu-driven program where only certain commands are available in each menu or to screen forms in a data entry program where certain words are used to fill in the form. They can also correspond to user tasks, such as spelling a word, that you'd like to keep distinct to help the recognizer. Sometimes a phrase will be translated differently in different contexts (the numeral 3 for spoken "three" in digit entry, the number word "three" for spoken "three" in text entry). Other phrases will be used only in one context and not in others.

The divisions of the vocabulary are called *subsets*. Subsets of the vocabulary are entered in separate columns of a language model. A language model can be a very simple table of just two columns. The first column shows what is to be said, the second what the KVS types when it recognizes the phrase in the first column. A language model with multiple subsets has 3 or more columns. Every subset consists logically of a column of inputs matched with a column of outputs.

A single language model may have up to 50 subsets, each with a unique name that you supply. The following rules apply to subset names:

1. Any printable characters from the ASCII set may be used. Internal spaces are not allowed in subset names.

2. Up to 25 characters are allowed.

3. A subset name must not begin with star (*) or sharp (#).

Language model subsetting offers three main benefits:

- Multiple meanings for single words
- Improved recognition accuracy
- User control over application flow

Multiple Meanings

Some programs expect the same phrase translated differently in different contexts, as in the preceding example of spoken "three."

Improved Accuracy

Subsetting improves recognition accuracy when a vocabulary includes potentially confusable pairs of phrases like these:

"date" and "eight"

"file" and "five"

"import" and "input"

"text" and "six"

If the confusable phrases are put in different subsets, one member of the pair is always eliminated as a candidate for recognition, and correct recognition of the other phrase becomes much easier.

User Control over Application Flow

You can give the user step-by-step control over a voice entry task by including a voice command that does nothing more than enter (or leave) a subset and confirm the action with a window message. This lets users change subsets (translations) by voice command when they want to.

Also, it's important to let a user know when something has gone wrong to allow correction of the mistake. When misspeaks and misrecognitions send the application unexpected inputs, window messages help inform the user.

When a language model has subsets, some method is required to select which subset the KVS is to listen for at any given time. The active subset is the one that contains all the phrases that are legal candidates for recognition at that point in the program. Switching from one subset to another is the purpose of the <subset {name}> command. This command can be typed at the keyboard, can be sent from a voice application program, or can be a voice command, such as "enter dates," which the language model translates into the KVS command <subset dates>.

Sharp Subsets: #edit and #voice_command

Sharp subsets let you give spoken commands to the KVS itself, using voice when the pop-up window is showing.

The KVS listens for the phrases in a sharp subset when all three of the following conditions are met:

1. The <param listen {ON/OFF}> switch is set to ON (the default).

2. A subset labeled #edit or #voice_command (or both) is in the current phrase book.

3. The KVS pop-up window is on the screen (and the user is not recording tokens).

#edit contains translations for phrases you might say to <edit>, and #voice_command contains translations for phrases you might say to any other KVS angle bracket command. When you type <edit>, the KVS translates spoken phrases from the #edit subset. When you type any other angle bracket command (except <profile> or <train>), the KVS translates spoken phrases from the #voice_command subset. The #voice_command subset is currently limited to translations for angle bracket command text only.

If the <param listen> switch is OFF, you cannot talk to the KVS through the window. When the switch is OFF, typing an open angle bracket shuts off recognition, effectively pausing the KVS while the command is being entered. If the switch is ON, you can mix typing and talking within the KVS window. Eliminating either of the sharp subsets turns off recognition in the context of the absent subset.

For examples of how to write the sharp subsets into a language model, see Table 8.2, "#edit," "#voice_command,"and Table 8.3, "add phrase."

HOW BIG CAN A LANGUAGE MODEL BE?

Space on board the KVS limits how much text there can be in a language model. Here are some of the limits:

- Number of characters in a translation cell: 498 (1-2-3 limits this to 240).
- Number of subsets: 50.
- Amount of space for storing unique translation strings: 65 Kbytes.
- Number of 498-character translation strings: 63.
- number of phrases: about 900, but the KVS can recognize up to 1,500 phrases. This is a function of the average word duration and the number of tokens to be trained (see example).

  Example: At an average word duration of .5 second, 2,700 tokens, roughly, fill up KVS main memory. At three tokens per vocabulary word, the language model can have 900 half-second phrases. At an average word duration of 1 second, there is room for half that: 1,350 tokens, or 450 phrases.

These figures presume a token resolution of 4 (the default). A higher resolution means more data are stored for each token and reduces the maximum number of phrases. A lower resolution means less data per token and therefore more phrases.

The limits can be pushed because the KVS will store additional tokens in a reserve memory, but when main token memory fills, response time lags slightly.

You can see how full memory is by going into training (<train .>) and typing [P] for pause.

CONSIDERATIONS FOR LANGUAGE MODEL DESIGN

Requirements of the Application

Translation strings for application-dependent vocabulary are determined by the keystrokes an application expects to receive. Some applications are case-sensitive to keyboard input, and therefore to the keyboard characters the KVS returns in translation strings. If the application requires all uppercase data, for example, you must enter the translation as all uppercase. It is up to you, the designer, to enter phrase translations in the exact format the host application program requires.

Ease of Use

Design your language model to achieve the following:

1. Give confirmation and feedback when a user says something.

   The system should always make some response when a user speaks, even if it only shows "no good match." This lets the user know immediately if he or she has been heard. **Example**: "mike off" translates to <microphone off> <window The microphone is off>.

2. Give users commands that they can use to get help and status information. (See Table 8.4.)

   Example: A command like "show vocab," or "where am I" in the PHRASE column, translates to <window Listening for digits>.

3. Give users command to correct misrecognitions and misspeaks and keep track of application flow. (See Tables 8.2-8.4.)

   Example: A command like "back one" in the PHRASE column translates to \b\s\b, backing up the cursor, overprinting the displayed

character with a space, and backspacing again to be ready for the new input.

4. Let the user run the KVS by voice command. (See Tables 8.1-8.4.)

The language model notation should be consistent to help a speaker remember what to type when editing and training a language model. Don't abbreviate (for example, use microphone off, not mike-off). Make subset labels informative (use part number, not subset 1).

KVS Requirements

To satisfy KVS performance requirements, design your language model to provide the following.

1. Distinctive vocabulary phrases

   Recognition accuracy is improved if you select vocabulary phrases that sound quite different from one another or if you relegate confusable phrases to different subsets. Words that rhyme or sound similar might be confusable and should be eliminated from the vocabulary set when possible.

   When this can't be done, the language model can provide another way to do the same thing. For example, if the KVS has trouble distinguishing the spoken numbers fifteen and nineteen, the end-user can say "one five" or "one nine." A language model can also provide synonyms. Synonyms are different ways to say the same thing, or words that give the same translation. A user could train both "nineteen" and "one nine" and later use whichever phrase works bests

2. Multiple Subsets

   Vocabulary for different application tasks should be partitioned into different subsets. Limiting the recognizer's choices to just the words in one subset improves accuracy.

   Multiple subsets also allow the same phrase to have multiple meanings: for instance, "three" can translate to the numeral 3 if the context is numeric -- calculation or data entry of numeric fields -- and the word "three" otherwise.

3. Phrases no longer than 2.0 seconds long

   The KVS is limited to recognizing 2.0 seconds of continuous speech. If a phrase is longer, break it into smaller units.

The next section summarizes star and sharp phrases (Tables 8.1 and 8.2) and sharp subsets (Table 8.2), basic voice commands (Table 8.3), and user help commands (Table 8.4). Examples of how to write a language model using the phrase follow each table.

TABLE 8.1 Star Phrases

PHRASE program name	KVS program function; when it executes	TRANSLATION STRING sample argument
*Enter	Executes the command listed as an argument after switching into this translation subset.	<window listening for dates>. The sample argument is a window message confirming that the KVS has started listening for a different set of words. <param wordgap 2>. This command shortens the pause required between words for this subset. Can be used to speed up digit entry if there is a digit subset.
*Exit	Executes the command listed as an argument before leaving this translation subset.	<param wordgap 5>. This command resets the pause length to normal duration.
*Unknown	Executes when the recognized word does not match any vocabulary in the phrase book. Translation usually puts up a window with a user help message.	<window PLEASE REPEAT> <beep>. This command makes the host beep whenever it does not recognize a word.
*Illegal	Executes when the recognized word is not in the current subset. The translation usually puts up a window with a user help message.	<window That word is not in Date subset>
*Suffix	Appends the translation string to all recognized words in the subset before they are sent to the host.	\s
*Prefix	Prefixes the translation string to all recognized words in the subset before they are sent to the host.	$
	Cancels affix for this word only	\\a

NAME: ***ENTER** -- The KVS executes the <command> and/or message that is listed as the translation of *ENTER when switching into that translation subset.

EXAMPLE OF USE:

To shorten the required pause between words and show the message "digit mode" when switching to a Digits subset.

COMMENTS:

Any KVS control commands listed as the translations for *ENTER are executed whenever the subset is entered.

When the KVS starts recognizing digits, a <window> command posts a message to the screen to inform the speaker that the KVS now expects to hear number words. A <param> command shortens the required pause between words. *ENTER translates to this sequence of commands in the Digits subset.

The window message is defined only once in the language model and is thus easier to update or modify. The subset can be entered from many points in an application, and the same message will always appear in the window. Notice that the <window> command is followed by a <!wait> command and then a second <window> command empty of text. <!wait> assures that the first window will stay on the screen a second and a half -- long enough to be seen before the second <window> command blanks it out. Without the second <window> command, a <window> message is not erased until the user presses a key.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	Digits
*ENTER		<param wordgap 2><window recognizing digits> <!wait 1500><window>
enter digits	<subset digits>	
one		1
two		2
.		.
.		.
nine		9
where am I		<window subset is digits><!wait 1500 ><window>

SEE ALSO:

*EXIT; Chapter 8: <SUBSET>; Appendix B: <WINDOW>.

NAME: *EXIT -- The KVS executes the (command) or <window message> listed as an argument for *EXIT when leaving this translation subset.

EXAMPLE OF USE:

When the KVS switches out of the Digits subset, the pause between words that had been shortened to speed up digit entry returns to its normal (default) value.

COMMENTS:

Any KVS control command(s) listed as the translation for *EXIT are executed when the KVS switches to a new subset and before the new subset is entered.

Different phrases can exit the subset, but because the *EXIT routines are defined only once, they are easy to update and modify.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	Digits
*ENTER	<window command mode>	<window digit mode> <param wordgap 2>
*EXIT		<param wordgap 5>
enter digits	<subset digits>	
enter commands		<subset commands>
one		one
two		two
.		.
.		.
nine		nine
where am I		<window subset is digits>

SEE ALSO:

*ENTER; Chapter 8: <SUBSET>; Appendix B: <WINDOW>.

NAME: ***UNKNOWN** -- The KVS executes the <command> or <window message> listed as a translation for *UNKNOWN whenever a spoken phrase scores below a rejection threshold. Most likely the phrase is not in the subset or the voacabulary; the KVS found no token that scored high enough to enter the candidate list.

EXAMPLE OF USE:

To enter a 2-digit number, you are required to say each number separately. If you say "fifteen" instead of "one five," the KVS beeps. The word "fifteen" is not in the PHRASE column. No token scores above the recognition threshold.

COMMENTS:

"fifteen" is not in the PHRASE column.

As in the example, a language model can give a different response depending on the application context. If a speaker is dictating a credit card number, the KVS can return a beep when the speaker says an unknown word. If the speaker is entering a command, the KVS can respond with a window message.

If a phrase is in the language model but the KVS reports it as *UNKNOWN, it is not scoring well enough to be recognized. The recognition threshold can be lowered to allow more uncertain matches.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Digits	Text
*UNKNOWN	<beep>	<window PLEASE REPEAT -unknown>
one	1	one
two	2	two
.	.	.
.	.	.
eight	8	eight
nine	9	nine

SEE ALSO:

*ILLEGAL; Appendix B: <threshold>.

NAME: ***ILLEGAL** -- The KVS executes the <command> or <window message> listed as a translation whenever a recognized phrase has no translation in that subset.

EXAMPLE OF USE:

You say "enter digits." The KVS switches to the Digits subset, and the KVS window shows "digit mode." Next you say "train mode," which has no translation in the Digits subset (see below). The host beeps; the KVS window shows "That word is not in the digit vocabulary."

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	Digits
*ILLEGAL	<beep>	<beep><window That word is not in the digit vocabulary>
*ENTER		<window digit mode>
one		one
two		two
.		.
.		.
nine		nine
enter digits	<subset digits>	
*enter		<window digit mode>
train mode	<train .>	

SEE ALSO:

*UNKNOWN; Chapter 8: <SUBSET>; Appendix B: <WINDOW>.

NAME: ***PREFIX**--The KVS adds the translation of *PREFIX to the beginning of every translation in that subset.

***SUFFIX** --The KVS adds the translation of *SUFFIX to the end of every translation in that subset.

\\a --The KVS will not add a prefix or suffix to a translation beginning with \\a.

EXAMPLE OF USE:

*SUFFIX and *PREFIX: When the KVS is recognizing from the Cost subset and you say "eighty-five," it returns $85.00.

COMMENTS:

An application requires every dollar amount entered to end with a decimal point and two zeros (.00). It also requires every dollar amount entered to begin with $. So .00 and $ are added to every entry in this subset except those beginning with \\a, which means "don't affix any characters to this string." \\a, no affix, cancels the *SUFFIX or *PREFIX command for that phrase only. Avoid repetitive affixes.

CAUTION: Always add \\a to translations that have only <commands> without text.

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Cost
*Suffix	.00
*Prefix	$
eighty-five	85
eighty-six	86
eighty-seven	87
quit	\\a 0\032\<window quit>

TABLE 8.2 Sharp Phrases

PHRASE program name	KVS program function; when it executes	TRANSLATION STRING sample argument
#delete last word	Executes when the phrase trained as #delete last word is recognized; removes the last word[1] from the screen one character at a time. The program executes as many times as there are characters in the last recognized word. The argument should be the keystroke or key sequence the host operating system interprets as "delete a key." The user can delete up to 10 previous words by repeating this phrase up to 10 times.	\b\s\b when [BACKSPACE] just moves back without erasing, or simply \b when [BACKSPACE] does both: moves back *and* erases
#next choice	Executes when the phrase trained as #next choice, is recognized; deletes the last word[1] typed, then displays the next choice recognition candidate.	\b\s\b,or \b
#try again	Executes when the phrase trained as #try again, is recognized; deletes the last word[1] typed and then listens for the user to say the word again. When the word is said again, the KVS will get it right. It will not make the same mistake again because it excludes the word just erased as a recognition candidate. Only the previous word can be retried.	\b\s\b or \b
#listen to me	Executes when the phrase trained as #listen to me is recognized; the KVS resumes sending translation strings to the host computer and executes any command listed as an argument. See <microphone OFF>.	<window mike is on>

1. *word* means the entire translation string. It can be a single word, a phrase, or a whole paragraph of up to 498 characters.

TABLE 8.2 (continued) Sharp Subsets

#SUBSET name	#SUBSET function; when it's used	TRANSLATIONS
#edit	When the <edit> command is given, the KVS automatically uses this subset for translation if it is present in the current phrase book *and* <param listen ON> is set. <param listen ON> enables the KVS to recognize speech when the pop-up window is on the screen. Any text that you can type into the window, you can speak instead.	All answers a user could make to select an item from an <edit> menu and respond to an <edit> prompt.
#voice__ command	When a user types an angle bracket command (other than <edit>), the KVS uses this subset for translation if it is present in the current phrase book *and* if <param listen ON> is set. However, the KVS will not recognize speech while you are recording tokens with <profile> and <train>.	All arguments or answers a user could type while entering an <angle bracket command>.

NAME: **#DELETE LAST WORD** -- The KVS removes the last recognized word from the screen when you say this phrase. To do this, it sends the translation of #DELETE LAST WORD as many times as there are characters in the word being deleted.

EXAMPLE OF USE:

You say "scratch that." The last word typed is erased from the screen. You can do this ten times. The KVS remembers the last ten phrases it sent to the host.

COMMENTS:

During training, when the screen prompted you to say #DELETE LAST WORD, you said "delete last word," or "erase" or "scratch that," or whatever phrase you wanted to use in recognition.

When it recognizes the phrase you trained for #DELETE LAST WORD, the KVS returns a string of \b -- backspaces -- one for each character to be deleted, and the last word typed is erased from the screen. That is, the KVS deletes as many characters as were in the last word typed, where *word* means the entire translation. -- a word, a phrase, or a paragraph of up to 498 characters. The string \b\s\b may be used for applications that don't erase when they backspace. The translation cell should contain the keystroke or key sequence that the application program interprets as "delete a key." Some applications use the [DELETE] key (CTRL-BACKSPACE, \177). Some simply use the [BACKSPACE] key (\b). KVTEST uses just BACKSPACE.

Some applications may not have appropriate backspace sequences. For example, if a translation string deleted a paragraph from a document, you cannot backspace to "undo" it.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	Digits
#DELETE LAST WORD	\b\s\b	

SEE ALSO:

#NEXT CHOICE: #TRY AGAIN

NAME: **#NEXT CHOICE** -- The KVS removes the last phrase typed from the screen and displays the next recognition candidate.

EXAMPLE OF USE:

You say "nine." The KVS recognizes "five." You say "next choice." "five" (the last phrase the KVS typed) is erased from the screen and replaced with the next best candidate. The second candidate is usually the phrase you want. If it isn't, you can say "next choice" again. You can repeat the "next choice" command until you have seen the entire candidate last. When there are no more possibilities, the phrase will be erased, but no other phrase will appear.

COMMENTS:

The translation should be the key or key sequence the host application program interprets as "delete a character." In response to #NEXT CHOICE, the KVS deletes as many characters as there are in the last phrase. The last phrase typed is erased from the screen. You can use #NEXT CHOICE to see alternatives only for the last recognized phrase.

#NEXT CHOICE erases a phrase just as #DELETE LAST WORD does. Then #NEXT CHOICE types the next best recognition candidate in its place. The KVS forgets alternatives when you say a new phrase.

NOTE

#NEXT CHOICE can be used only in the default output mode, top_choice, as set with the command <output top_choice typeahead>.

LANGUAGE MODEL EXAMPLE

A	B	C
PHRASE	Commands	Digits
#NEXT CHOICE		\b\s\b
one		1
two		2
five		5
eight		8
.	.	.
.	.	.
nine		9

SEE ALSO:

#DELETE LAST WORD; #TRY AGAIN.

NAME: **#TRY AGAIN** -- The KVS removes the last phrase typed from the screen. The user says the phrase again. The KVS will not make the same mistake again; it recognizes the new phrase, omitting the one erased as a recognition candidate.

EXAMPLE OF USE:

You say "nine." The KVS recognizes "five." You say "try again nine." The phrase "five" is erased from screen, and the KVS recognizes "nine." No tokens of "five" are allowed to enter the candidate list.

COMMENTS:

Use #TRY AGAIN when the KVS confuses two phrases that sound alike. After saying "try again," you must also say the phrase to be recognized.

The translation of #TRY AGAIN should be the key or key sequence that the host application interprets as "delete a character." (See #DELETE LAST WORD.)

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	Digits
#TRY AGAIN		\b\s\b

SEE ALSO:

#DELETE LAST WORD; #NEXT CHOICE.

NAME: **#LISTEN TO ME** -- The KVS turns the microphone back on and also executes any <command> or <window message> if one is listed as the translation.

EXAMPLE OF USE:

#LISTEN TO ME triggers <microphone ON>, the second of a sequence of two commands that turn the microphone off and then back on while you are recognizing. The first command, <microphone OFF> causes the KVS to recognize only the phrase you trained as #LISTEN TO ME.

<microphone ON> is the implied translation of the sharp phrase #LISTEN TO ME. You do not have to supply the translation in the language model. This translation works automatically in all subsets.

COMMENTS:

A good translation for "listen to me" is a <window> command telling the user that the microphone is on.

The <microphone OFF> command causes the KVS to listen for only one phrase: the one you have trained as #LISTEN TO ME. The <pause> command, by contrast, just turns the microphone off. #LISTEN TO ME will not work after <pause> has been used to turn the microphone off.

You can also issue these commands from a voice application program and turn the microphone on and off under program control.

In the #EDIT subset, the translation for "microphone off" must be \026 [CTRL-V], not <microphone off>. #LISTEN TO ME needs no translation in the #EDIT subset.

LANGUAGE MODEL EXAMPLE:

A	B	C	D
PHRASE	Commands	Digits	#EDIT
microphone off #listen to me	<microphone off> <window I'm listening>	<microphone off> <window I'm listening>	\026

SEE ALSO:

Appendix B: <microphone>, <pause>, <window>.

NAME: **#EDIT** -- The KVS uses a special recognition subset for the <edit> command.

EXAMPLE OF USE:

While recognizing, you want to add a new phrase. You say "add phrase," and the <edit .> function appears in the KVS window. Using the spoken letters or a phonetic alphabet, you spell the phrase to be added, and spell its translations in one or more subsets. Then you say "train mode." The <train function> appears in the KVS window prompting you to say the phrase, and you're back in recognition running the application, all by voice command. (Any part of this process can be entered from the keyboard instead.)

COMMENTS:

The KVS automatically translates spoken phrases from a special subset labeled #EDIT whenever a user types the <edit> command or says a phrase that is translated as <edit>. When a user leaves <edit> to train or to recognize, the KVS automatically switches back to the subset it was using before entering the #EDIT subset. The #EDIT subset should contain all phrases and translations a user might need to run <edit> by voice. (See also Chapter 12.)

This example includes the phonetic alphabet so that a user can spell a phrase to be added and the phrase "train mode" so the user can escape from the "add phrase" function and go from <edit> into <train>. <PARAM listen ON> must be set to use voice commands in <edit>.

LANGUAGE MODEL EXAMPLE:

A	B	
PHRASE	#EDIT	reflex
Before adding a phrase:		
edit text	\032e	<edit .>
add phrase	\032a	<!choose a><edit .>
alpha	a	
bravo	b	
charley	c	
delta	d	
echo	e	
.		
.		
yankee	y	
zulu	z	
train mode	\032\r	<train .>
return	\r	
escape	\e	

(*continued on next page*)

LANGUAGE MODEL EXAMPLE (continued):

A	B	
PHRASE	#EDIT	reflex

After adding a phrase:

banana	banana

SEE ALSO:

#voice_command; Appendix B: <edit>, <choose>, <param listen>.

NAME: **#VOICE_COMMAND** -- The KVS uses a special recognition subset for <angle bracket> commands other than <edit>.

EXAMPLE OF USE:

This example illustrates the use of the #VOICE_COMMAND subset to edit tokens by voice. While recognizing, you want to add or replace tokens of phrases. You say "train tokens," and the KVS window appears prompting you to say either "add" or "replace token," spell out the phrase with the phonetic alphabet and then say "training." The prompt stays on the screen briefly and then the window changes to a choose command that looks like this:

<!choose

When you say "add" or "replace tokens" and spell the phrase, [a] or [r] and the phrase are added to the choose command. For example

<!choose afive

is an argument to add tokens of the phrase "five." When the choose command is correct, you say "training." The KVS automatically completes the !choose command with a >, and the window switches to training. The train menu appears briefly and then the prompt to say the phrase to which you are adding or replacing tokens. When token editing is complete, the KVS automatically switches back to recognition using the subset that was current before you said "train tokens."

COMMENTS:

The KVS automatically translates spoken phrases from a special subset labeled #VOICE_COMMAND whenever a user types any <angle bracket> command (except <edit>) or says a phrase that is translated as an <angle bracket> command. When the command completes, the KVS automatically switches back to the subset it was using before entering the #VOICE_COMMAND subset. The #VOICE_COMMAND subset should contain all phrases and translations a user might need to run <angle bracket> commands by voice. You can make corrections to the window by saying any of the sharp phrases for voice correction: #DELETE LAST WORD, #TRY AGAIN, or #NEXT CHOICE.

This example includes the phonetic alphabet so that a user can spell a phrase whose tokens are to be edited and the phrase "training" so a user can initiate and escape from "train tokens." <PARAM listen ON> must be set if you want to edit tokens by voice.

(*continued on next page*)

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	#VOICE_COMMAND
train tokens	<WINDOW Say "ADD TOKEN" or "REPLACE TOKEN" \n then spell out the phrase with the phonetic alphabet\n then say "training"><!wait 4000><!choose
add token	a
replace token	r
training	\\r\\r\r<train .>
alpha	a
bravo	b
charley	c
delta	d
echo	e
.	
.	
yankee	y
zulu	z
return	\r
escape	\e

SEE ALSO:

#EDIT; "add token," "replace token;" Appendix B: <train>, <choose>, <param listen>; Chapter 10.

TABLE 8.3 -- Voice Commands and Translations

Spoken Phrase	Translation	Voice Command Function
"add phrase"	<!choose a><edit .>	Calls <edit .> to add a phrase to current phrase book.
"add tokens"	<subset add_tokens>	Exits recognition into training and adds 3 tokens of a phrase.
"replace tokens"	<subset replace_tokens>	Exits recognition into training and replaces all tokens of a phrase.
"train tokens"	see #voice_command	Exits recognition, then adds or replaces all tokens of a phrase and returns automatically to using the subset that was current before the user said "train tokens"
"test recognition"	<output ordered_list typeahead><recognize .>	Tests recognition of an untried language model after training.
"microphone check"	<recognize .>	Starts a microphone check or a hands-off microphone check.
"microphone off"	<microphone off >	Turns off the microphone until you say #LISTEN TO ME.
"back three"	\b\s\b\b\s\b\b\s\b	Deletes the last 3 characters from the screen.
"set threshold low"	<threshold 3>	Can improve recognition if the KVS appears deaf.
"set threshold high"	<threshold 6>	Can improve recognition when the KVS is giving incorrect answers due to high background noise.
"quit"	\032 <window quit>	Quits and shows what you say when the KVS sends a [CTRL-Z] to terminate KVTEST.
""train mode"	<train .>	Starts training.
"pause"	<pause>	Suspends KVS recognition.
"paragraph"	\r\r\t	Gives automatic paragraphing: two carriage returns and an indent.

TABLE 8.3 (cont.) Voice Commands and Translations

Spoken Phrase	Translation	Voice Command Function
"period"	\b.\s\s<subset caps>	Starts a new sentence and capitalizes the first word.
"standard close"	\t\t\t Sincerely yours,\r\r\r\r\t\t\tFred Spone, Manager\r	Types a boilerplate paragraph.
"lotus"	cd\\123\r123r<! wait 2000>\r	Runs several commands with a single phrase (executes a *voice macro*).
"return" or "process" or "enter" or "new line"	\r \r \r \r	Allows synonyms: alternative voice commands that accomplish the same thing.
"January"	Jan<subset quantity>	Types a field and automatically starts listening for a different set of words *(branches)*.
"delete all"	<subset confirm>	Protects against accidentally losing data.
"four"	4/for	Types all homonyms. The wrong one can be deleted in a later editing pass.
"outside line"	ATDT<!wait 2000>893-5151\r	Requests the modem to go "off hook," waits two seconds and then sends it a phone number to dial.

NAME: **"add phrase"** -- Invokes <edit> to add a phrase to your vocabulary.

EXAMPLE OF USE:

You're testing a language model and discover you'd like to add a new phrase. You say "add phrase." The KVS calls the <edit> command, chooses the [A] (add phrase) function, and pages to the next screen so you can type or spell by voice the phrase to add. After typing or spelling the phrase, you can type or say the translation, go on to add another phrase, or train the phrase you just added.

The #EDIT subset in the following example lets you spell "banana," select a translation subset via CTRL-S (\023), enter a translation string that is the same as the phrase via the TAB key (\t), return to the main <edit> menu, and then train the phrase you just added.

COMMENTS:

The <!choose> command lets you define the keystrokes to type for any KVS command and feeds the keystrokes to the command one by one as run-time arguments.

<!choose a ><edit .> transfers control to the <edit> command and then sends "a" for "add phrase." Once <edit> has control, the KVS automatically translates from the #EDIT subset if it is present so you can edit a language model by voice.

You can mix talking and typing within <edit>, entering new phrases and control commands by spelling them or speaking them. You can also turn the microphone off and on again by voice.

<!choose> translations for running <edit> can be cryptic. Before entering a <!choose> translation string in your language model, go through the <edit> sequence manually; write down every keystroke you need and then enter it in the language model.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	#EDIT
Before adding a phrase:		
add phrase	<!choose a><edit .>	
alpha		a
bravo		b
charley		c

(continued on next page)

LANGUAGE MODEL EXAMPLE (continued):

A	B	C
PHRASE	Commands	#EDIT
delta		d
.		
.		
train mode		\032\r
return	\r	
subset		\023
repeat entry		\t
mike off		\026
#listen to me		\026\026
After adding a phrase:		
banana	banana	

SEE ALSO:

#EDIT; Chapters 10 and 12; Appendix B: <edit>, <choose> <train>.

NAME: **"add tokens"** -- The KVS exits recognition, enters training, and adds three tokens of the specified phrase.

EXAMPLE OF USE:

Adding tokens to any phrase in the language model.

COMMENTS:

This language model example provides hands-off addition of tokens: all necessary keystrokes are delivered by the translation strings in the subset add__tokens.

When the user says "add tokens," the KVS switches to the subset of the same name. Entering the subset prompts the user with the window message "Say the phrase to add tokens to."

When the user says the phrase (all phrases in the language model are included in the subset), the KVS assembles the characters that spell the phrase with the keystrokes necessary to add tokens to the phrase.

As an example, if the user says "one," the KVS sends the following translation string:

<!choose aone\\r\\r><train.>

This sequence of commands and keystrokes puts the KVS in train mode, displays the <train> menu, automatically presses [a] for add tokens, types "one" (the phrase to add tokens to), enters it with [RET] and then starts up token recording. Next the <train> command prompts for three tokens of "one." When the user has said three tokens, the <train> menu returns and the <!choose command uses the second [RET] to exit training into recognition.

Notice the use of *PREFIX and *SUFFIX to pass the keystrokes required to choose [a] from the <train> menu and to press [RET]. Notice also that <!choose is an angle bracket command without a terminator. The characters that spell the recognized phrase and two carriage returns are appended before *SUFFIX terminates <!choose>.

LANGUAGE MODEL EXAMPLE: (*continued on next page*)

LANGUAGE MODEL EXAMPLE :

A	B	C
PHRASE	all	add_tokens
add tokens	<subset add_tokens>	
*ENTER		<window Say phrase to add tokens to>
*PREFIX		<!choose a
*SUFFIX		\\r\\r><train .><subset all>
one	1	one
two	2	two
three	3	three
.		.
.		.
nine	9	nine
where am I	<window subset all>	<window subset add_tokens>
*ILLEGAL		<window phrase is not allowed now>
*UNKNOWN	<window phrase unknown>	<window phrase unknown>

SEE ALSO:

#VOICE_COMMAND; "replace tokens"; Appendix B: <choose>, <train>; Chapter 10, "Adding, Replacing and Deleting Tokens."

NAME: **"replace tokens"** -- The KVS exits recognition, enters training, and replaces a token of the specified phrase.

EXAMPLE OF USE:

Replacing all tokens of any phrase in the language model.

COMMENTS:

This language model example provides hands-off replacement of tokens. All necessary keystrokes are delivered by the translation strings in subset replace_token.

When the user says "replace tokens," the KVS switches to the subset of the same name. Entering the subset prompts the user with the window message "Say the phrase to replace."

When the user says the phrase (all phrases in the language model are included in the subset), the KVS assembles the characters that spell the phrase with the keystrokes necessary to replace all tokens of the phrase.

As an example, if the user says "one," the full translation string the KVS assembles is as follows:

<!choose rone\\r\\r><train .>

This sequence of commands puts the KVS in train mode, displays the <train> menu, automatically presses [r] to replace tokens from the menu, types "one" (the phrase to replace), enters it with [RET] and then starts up token recording. Next <train> prompts for as many tokens of "one" as currently exist in the token set. When the user has said all tokens, the <train> menu returns, and the <!choose string uses the second [RET] to exit training into recognition.

Notice the use of *PREFIX and *SUFFIX to select [r] from the menu and press [RET]. Notice also that the <!choose r translation string is an open-ended angle bracket command. The characters that spell the recognized phrase and two carriage returns are appended before *SUFFIX terminates <!choose>.

LANGUAGE MODEL EXAMPLE: *(continued on next page)*

LANGUAGE MODEL EXAMPLE (continued):

A	B	C
PHRASE	all	replace_token
replace tokens	<subset replace_token>	
*ENTER		<window Say phrase to replace>
*PREFIX		<!choose r
*SUFFIX		\\r\\r><train .><subset all>
one	1	one
two	2	two
three	3	three
.		.
.		.
nine	9	nine
where am I	<window subset all>	<window subset replace_token>
*ILLEGAL		<window phrase is not allowed now>
*UNKNOWN	<window phrase unknown>	<window phrase unknown>

SEE ALSO:

#VOICE_COMMAND; "add tokens"; Appendix B: <choose>, <train>; Chapter 10, "Adding, Replacing and Deleting Tokens."

NAME: **"test recognition"** -- The KVS starts recognition in ordered output mode.

EXAMPLE OF USE:

To test recognition after training an untried language model.

COMMENTS:

This command puts the KVS in ordered output mode and starts recognition with a hands-off microphone check.

The <output ordered_list typeahead> command specifies a format and destination for KVS recognition results. The format is a list of the top-scoring phrases, from highest to lowest. There are five or fewer phrases in the list by default. A look at the list will show you how well you have trained the KVS and whether the KVS confuses any pairs of phrases in the language model. Phrases that sound alike will show up in the output list with closely scored tokens. Such phrases should be changed for good recognition performance. Chapter 11 tells you how to analyze recognition results and what action to take when you turn up confusable phrases.

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Commands
test recognition	<output o t><!choose \\r\\r\\r><recognize .>

SEE ALSO:

Appendix B: <param maxphrases>, <output>; Chapter 11, "Find Problem Words"; Chapter 13, "Adjusting Parameters Controlling Recognition."

NAME: **"microphone check"** -- The KVS begins recognition with a microphone check.

EXAMPLE OF USE:

The KVS is recognizing. You say "mike check,. The KVS window shows

```
press   [RET] to begin microphone check,
        [ESC] to escape,
        [G]   to set microphone gain.
```

COMMENTS:

This voice command assumes that recognition is already on. It puts the KVS into recognition mode with the currently loaded phrase book; automatically starting a microphone check.

The second example below shows a hands-off microphone check requiring no keyboard entry. It bypasses the gain setting, enters the check of background noise, and then prompts for the test sentence,"Testing one two..." (Gain can be set using a <!choose> command too.)

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	Hands-off
micropohone check	<recognize.>	<!choose \\r\\r\\r><recognize .>
hands off	<subset hands_off>	

SEE ALSO:

Appendix B: <pause>.

NAME: **"microphone off"** -- The KVS stops listening for anything but "listen to me."

EXAMPLE OF USE:

You say "microphone off." The KVS window shows: Microphone is off. The KVS stops typing. Say "Listen to me" to resume.

COMMENTS:

In the #EDIT subset, the translation of "microphone off" is \026 instead of <microphone off>.

LANGUAGE MODEL EXAMPLE

A	B
PHRASE	Digits
*ENTER	<window digit mode>
microphone off	<microphone off><window Microphone is off. Say "Listen to me" to resume>
#LISTEN TO ME	<window Microphone is on>

SEE ALSO:

#LISTEN TO ME; Appendix B: <microphone>, <pause>.

NAME: **"back three"** -- The KVS removes the last 3 characters from the screen.

EXAMPLE OF USE:

You say "back three." The last 3 characters are erased from the screen.

COMMENTS:

You can delete single characters instead of phrases by writing such phrases as "back one" or "back three" in your language model. Include the string \b\s\b once for each character to be deleted. This is useful for programs in which [BACKSPACE] moves the cursor without deleting the character. \b\s\b will, for some applications delete the character as well as backspace.

If your application program uses the [BACKSPACE] key to erase, you need only enter \b. You may find that the [DEL] key or [CTRL-BACKSPACE] is used instead. The "delete a character" function is application-specific.

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Commands
back one	\b\s\b
back three	\b\s\b\b\s\b\b\s\b

SEE ALSO:

#DELETE LAST WORD; #NEXT CHOICE; #TRY AGAIN.

NAME: **"set threshold low"** -- The KVS lowers the rejection threshold.

EXAMPLE OF USE:

Background conditions are noisy or you are using someone else's training. You've just done a microphone check, but the KVS acts as if it can't hear you. You say "set threshold low," and the KVS acknowledges you with a window message. Recognition improves. (If the KVS can't hear you, type the <threshold> command to lower the rejection threshold.)

COMMENTS:

Resetting the rejection threshold adjusts the BSCORE parameter. BSCORE is the minimum score a token must have before it can enter the candidate set. Lowering the <threshold> by 1 lowers BSCORE by 50. A low threshold admits more questionable recognition matches into the candidate set. This example resets the threshold to 3, a BSCORE of 400. Threshold defaults to a value of 5; BSCORE defaults to 500. Notice that the command to lower the threshold is available in every subset because the translation is entered in each column.

Remember that lowering the recognition threshold allows the KVS to propose poorer matches as recognitions. After changing the <threshold>, be sure to test recognition. You may have to raise the <threshold> again or adjust BSCORE to get back to acceptable performance.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Commands	Digits
set threshold low	<thresh 3>	<thresh 3><window rejection threshold is 3>
set threshold high	<thresh 5>	<thresh 5><window rejection threshold is 5>

SEE ALSO:

"microphone check"; Chapter 13; Appendix B: <parameter bscore>, <threshold>.

NAME: **"quit"** -- The KVS window displays what you say when the KVS sends a nonprinting [CTRL-Z].

EXAMPLE OF USE:

To exit the KVS voice application program, KVTEST, you must type [CTRL-Z]. The ASCII code for [CTRL-Z] is \032, a nonprinting code. Instead of typing [CTRL-Z], you say "quit." The KVS sends the nonprinting code to the host PC, and "quit" is displayed in the KVS window.

COMMENTS:

Any nonprinting key sequence should be accompanied by a message. The message is displayed in the window using the <window {message}> command, where {message} is a verbal equivalent of the key sequence. The KVS sends the nonprinting key to the PC host and, because the user sees no change on the PC screen, the window pops up to let the user know which key was sent.

If the language model puts up the KVS window when it recognizes "quit,"the speaker can see what was said. Consider the situation where a speaker says "pit" and, mysteriously, the application program exits. The speaker will be confused unless there is a window message making it evident that the utterance was misheard. With visual feedback, the speaker can correct the situation and retrain "quit" so the stored tokens don't sound like "pit."

NOTE

If the phrase "quit" is confusable with other phrases in the language model, try "exit" or "end of session."

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Commands
"quit"	\032 <window quitting>

SEE ALSO:

Appendix B: <window>, <beep>.

NAME: **"train mode"** -- Start training by voice.

EXAMPLE OF USE:

You are running an application, entering numbers in a data base, when you see that the KVS is confusing "nine" and "five." You decide to add more tokens of each phrase to improve recognition. You say "train mode" and the KVS window comes up with the training menu, allowing you to add tokens.

COMMENTS:

To control KVS recognition mode by voice command, your language model should include phrases that translate to KVS angle bracket commands. For example, you can pause and resume recognition, initiate training, edit language models. and set the recognition threshold, the output mode, or recognition parameters, all by voice.

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Commands
train mode	<train.>
pause	<pause>
top choice	<output top_phrase typeahead>
confirm mode	<output confirm typeahead>
set threshold low	<threshold 3>
set threshold high	<threshold 7>
edit mode	<edit.>
save phrase book	<!choose fs\r\032\r><edit.>
status	<status>
restart	<restart>
sign off	<bye>
add tokens	<!choose a><train.>
replace tokens	<!choose r><train.>

SEE ALSO:

Appendix B: "Angle Bracket Commands"; KVS Reference Manual.

NAME: **"paragraph"** -- Gives voice entry of new paragraph command.

EXAMPLE OF USE:

You can punctuate and format text with single commands that translate into a series of keystrokes. See comments and example below.

COMMENTS:

The "paragraph" command types two carriage returns and tabs in.

The "period" command starts a new sentence by typing a backspace, a period, and two spaces and switching to a subset where the first letter of each word is capitalized.

The "comma" command inserts a comma by typing a backspace, a comma, and a space.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Text	Caps
*SUFFIX		\s<subset text>
paragraph	\r\rt	
period	\b.s\s\s<subset caps>	
comma	\b,\s	
colon	\b:\s	
the	the\s	The
is	is\s	Is
this	this\s	This
Kurzweil	Kurzweil\s	Kurzweil
Plural	\bs\s	
Possessive	\b's\s	

SEE ALSO:

Appendix A: Backslash characters; Appendix B; Chapter 8: <subset>.

NAME: **"sincerely"** -- The KVS types a boilerplate paragraph.

EXAMPLE OF USE:

You're using a word processor to write a note. At the conclusion, you say "standard close." The KVS types a standard closing. It tabs in three times, types "Sincerely yours," gives four carriage returns, tabs in again and types your name, your title, and a carriage return.

COMMENTS:

Since translation strings can be up to 498 characters, a speaker can call up a whole paragraph with a single phrase. It is usually better to use two-word phrases for commands, such as "standard close." They are easier for the KVS to recognize reliably and allow recognition of single words as text.

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Text
standard close	\t\t\t Sincerely yours, \r\r\r\r\t\t\t Fred Spone, Manager\r

SEE ALSO:

Appendix A: "IBM PC Key Codes."

NAME: "lotus" -- The KVS runs several commands with a single phrase (executes a **voice macro**).

EXAMPLE OF USE:

You're using MS-DOS, and you want to use a spreadsheet to build a language model. You say "lotus." The KVS types the following MS-DOS and lotus commands:

c>cd\123[RET]
c>123\ 123 [RET]
[in 123] [ret]
[in 123] /fr

The first command changes to a directory called 123. The second command invokes the Lotus 1-2-3 spreadsheet program. The Lotus commands load a spreadsheet file.

COMMENTS:

These examples presume a fixed disk with a directory called 123 containing Lotus 1-2-3. Notice that the ability to run several commands with a single phrase is a macro capability. You can write voice macros for any application or operating system by stringing together the commands that are required to complete a procedure.

As another example, suppose in cleaning up your fixed disk you want to move all Wordstar backup files into a special directory called Archive. To complete this task you want to copy the .bak files, delete them from the current directory, and confirm the deletion by listing the directory contents when you are done. With the single spoken phrase "archive" you could run all the MS-DOS commands required to complete the task. See the example below.

LANGUAGE MODEL EXAMPLE:

A B

PHRASE	Commands
lotus	cd\\123\r123\r<!wait 2000>\r
load file	\fr
archive	copy *.bak\s\\archive*.*\rdel\s*.bak\rdir/w\r

NAME: **"return" or "enter" or "process"** -- The KVS types a carriage return.

EXAMPLE OF USE:

To enter most operating system commands, you say the command, the names of its arguments, and then you say "enter" or "return." The command is then executed.

To add information to a manufacturing data base, you fill out a screen form and say "process." The program updates your data base. All three commands do the same thing: they type a carriage return.

COMMENTS:

The possibility of adding synonyms brings a new flexibility to the strict rules for input that most computer programs depend on. Providing synonyms in the language model lets a speaker associate the same keystroke with a task-oriented phrase. It also lets different speakers choose the phrases they feel most comfortable with, and it avoids occasional problems with the recognizer for confusable phrases or for certain speakers.

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Commands
process	\r
enter	\r
return	\r
new line	\r
ok	\r

NAME: "January" -- The KVS types a field and automatically starts listening for a different set of phrases. The language model **branches** to a new subset after each recognition.

EXAMPLE OF USE:

You are adding employees to a personnel data base, filling out a screen form for each employee who has just joined the company. You are going to enter information in a fixed order. Each time you fill in a blank in the form, you will use a new group of phrases. You say "January" to fill in the month the employee joined the company. Your next field is "marital status." You'll speak one of three words. You want the KVS to stop listening for the names of the months and start listening for one of the three words defining marital status.

COMMENTS:

The following example shows that a single translation can contain both typing and angle bracket commands. The KVS will type "January" and switch subsets in anticipation of hearing a new set of phrases.

LANGUAGE MODEL EXAMPLE:

A	B	C	D
PHRASE	months	m-status	exemptions
*ENTER	<window say month>	<window speak status>	<window please enter exemptions>
january	1\r<subset m-status>		
february	2\r<subset m-status>		
march	3\r<subset m-status>		
married		m\r<subset exemptions>	
single		s\r<subset exemptions>	
divorced		d\r<subset exemptions>	
one			1\r
two			2\r
three			3\r

SEE ALSO:

*SUFFIX; Appendix B: <subset>; Chapter 13, "Remove Confusion with Subsetting."

NAME: "delete all" (**confirm**) -- Switches to a subset requiring you to confirm critical transactions. You say the command twice; this protects you from accidentally losing data.

EXAMPLE OF USE:

A voice command may be set up to delete all phrase books with a single phrase. Before translating and acting on this command, the language model requires user confirmation.

As another example, the format command within MS-DOS can reformat the PC fixed disk. Because reformatting erases so much data, a language model may require user confirmation before returning that command to MS-DOS.

COMMENTS:

Any command that causes an irrevocable action, especially one that destroys data, should require user confirmation on recognition. This protects against misrecognitions and misspeaking.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Main	Confirm
*ENTER		<window Say command again to confirm; say anything else to cancel>
*ILLEGAL		<subset main><window NOT DELETED>
*UNKNOWN		<subset main><window NOT DELETED>
delete<subset confirm>	delete\r<subset main><window DELETED>	

SEE ALSO:

*ILLEGAL; *UNKNOWN; Chapter 8: <SUBSET>; Appendix B: <WINDOW>.

NAME: "outside line" -- The KVS waits two seconds for an outside line before sending a modem a phone number to dial.

EXAMPLE OF USE:

You're controlling a program such as Crosstalk by voice. You request an outside line from the modem and then have it dial a number. To avoid sending the modem the number before it's ready to use it, you introduce a delay with the **<!wait>** command before sending the number.

COMMENTS:

<!wait> causes the KVS to delay before sending the next message or command. The delay is measured in milliseconds (thousandths of a second), where 1 second is 1000 msec. The duration of the delay is unlimited.

LANGUAGE MODEL EXAMPLE:

A	B
PHRASE	Commands
outside line	ATDT<!wait 2000>893-5151\r

SEE ALSO:

Appendix B: <!wait>; Chapter 11, "Delay KVS Message Transmission."

TABLE 8.4 User Help Commands and Translations

Phrase	Translation	Voice Command Function
"where am I"	<subset>	Display the name of the current translation subset.
"phrase books"	<recog ?>	Show available phrase books for recognition.
	<edit ?>	Show available phrase books for editing.
	<train ?>	Show available phrase books for training.
"show language model"	<!choose v t\027\r\027\r\027\r><edit .>	List language model for current phrase book: all phrases, subsets and translations. (Speaker presses [RET] or says "return" to page through the listing.)
"show vocab"	<!choose v p \027><edit .>	List vocabulary for current phrase book: all phrases. (Speaker presses [RET] or says "return" to page through the listing.)
"show subsets"	<subset?>	List available subsets for current phrase book.
"show language model size"	<!choose i><edit .>	Show language model size for current phrase book.
"show set-up"	<!choose s><edit .>	Show tokens /phrase, resolution, and voice profile name for current phrase book.
"show users"	<user?>	Show all names in \KVSUSERS.
"show commands"	<help>	List available angle bracket commands.
"show status"	<!status>	Display current user and phrase book and command in process if any.
"show datapath"	<datapath>	Display path to directory containing your phrase books.

NAME: **"where am I"** -- Display name of the current subset.

EXAMPLE OF USE:

You are using the newdates language model. You say "subset command" and then "KVTEST." Then you say "April." There is no response. You say "where am I?" The KVS window appears with the message "The current subset is command." This is a reminder that you haven't switched the KVS to the months subset yet, and so the KVS is only listening for commands. You say "set date". The KVS window appears, confirming that the KVS is now listening for dates. Now when you say "April," the host types 4- on the screen.

Notice that the language model is designed to leave *ENTER window messages on the screen for 1.5 seconds and then blank them out automatically.

COMMENTS:

Whenever a user has to rely on memory to use an application, provide help messages. See Table 8.4 for voice help commands.

LANGUAGE MODEL EXAMPLE:

A	B	C
PHRASE	Command	Months
*ENTER	<window command mode><!wait 1500><window>	<window dates mode><!wait 1500><window>
subset command		<subset commands>
KVTEST	KVTEST\r	
set dates	<subset months>	
where am I	<subset>	<subset>
March		3-
April		3-

SEE ALSO:

*ILLEGAL; *UNKNOWN; Appendix B: <window>, <subset>, <!wait>.

CHAPTER 9
MAKING A PHRASE BOOK

CONTENTS:

- The Files in a Phrase Book: Language Model, Voice Profile, Tokens
- How to Make a Phrase Book Starting with <edit>
- KVMAKELM: How to Make a Phrase Book Starting with a Spreadsheet Program
- KVLISTLM: How to Print a Language Model
- KVLISTLM: How to Return a Language Model to ASCII Format for Editing With a Spreadsheet
- KVMAKEPB: How to Put an Untrained Phrase Book in the End-User's Directory
- KVMODPB: How to Use Existing Tokens with a New Version of a Language Model

This chapter tells you how to use the KVS Host Utilities to

- print a language model (KVLISTLM);
- take changes you added to your language model with <edit> and put them back into spreadsheet format (KVLISTLM);
- put a new (untrained) phrase book directly in an end-user's directory without having to download and upload files between the PC and the KVS over the communications line (KVMAKEPB);
- link existing tokens with a modified ASCII language model (KVMODPB).

It also shows what files result when you make a phrase book with the <edit> command or with a spreadsheet program.

THE FILES IN A PHRASE BOOK: Language Model, Voice Profile, Tokens

A phrase book (diagrammed in Chapter 6, Figure 6.1) consists of a language model (phrases and their translations), your voice profile (the mathematical model of your voice), and tokens (samples of your speech for each phrase).

When you use the KVS to recognize and run an application program, the phrase book for that application must be present in your directory. The phrase book is downloaded into the KVS for recognition.

For each part of a phrase book -- language model, voice profile, and tokens -- Tables 9.1 and 9.2 tell you what programs produce them and what files they consist of. Any listed object can be made only after you have made the objects listed above it.

HOW TO MAKE A PHRASE BOOK STARTING WITH <EDIT>

Table 9.1 shows the files created by making a phrase book from scratch with the KVS <edit> command. If you compare this table with Table 9.2, you'll see that <edit> saves several steps initially. <edit> lets you work in your \KVSUSERS directory. The KVMAKELM utility is not needed.

TABLE 9.1 MAKING A PHRASE BOOK USING THE <edit> COMMAND

To make	From	Use	Result
A voice profile	recording tokens	KVENROLL	.kam, .kpm, .kvc and .kpl files stored in \KVSUSERS\{your name}
A untrained phrase book (1)	penciled notes and your voice profile	<edit> [S](save){pbook}	{pbook}.klm, .kmp, uploaded from the KVS and stored in \KVSUSERS\{your name}
A trained phrase book ....(1)	an untrained phrase book	<train> [S](save){pbook}	.kvc file added to the {pbook}
A backup copy of the phrase book	your enroll.kam enroll.kpm enroll.kpl {pbook}.klm {pbook}.kmp {pbook}.kvc	KVSMOVE/	Copies of these 6 files stored on a floppy disk in backup format

.kam: Acoustic model spanning sounds a speaker makes; one of two parts of a voice profile.
.kpm: Phonetic model capturing distinctions between speech sounds; one of two parts of a voice profile.
.kvc: Reference tokens.
.kpl: ENROLL log file; records which parts of enrollment are finished.
.klm: Language model with phrase-translation pairs in subsets.
.kmp: Phrase map describing prompting order and token locations; one of two parts of a language model.
\KVSUSERS: Main directory containing a subdirectory for each enrolled user; created by INSTALL.
{your name}: The name you supplied when you signed on with <user {your name}>.
{pbook}: Application phrase book name.

(1) See note to Table 9.2.

Table 9.2 describes the creation of phrase book files when you work with a language model in ASCII format.

TABLE 9.2 -- MAKING A PHRASE BOOK FROM AN ASCII FILE

To Make	From	Use	Result
An ASCII format language model	penciled notes or ideas	a spreadsheet or text editor	a .prn file
A KVS format language model	a .prn file	KVMAKELM	.klm and .kmp files created and stored in \KVSLMOD
A voice profile	reference tokens	KVENROLL	.kam, .kpm, .kvc and .kpl files stored in \KVSUSERS\{your name}
A trained phrase book (1)	language model, voice profile and training	<train> [S](save){pbook}	.klm .kmp and .kvc files uploaded from the KVS and stored in\KVSUSERS\{your name}
A backup copy of the phrase book	your enroll.kam enroll.kpm, enroll.kpl {pbook}.klm {pbook}.kmp {pbook}.kvc	KVSMOVE/	Copies of these 6 files stored on a floppy disk in backup format

.prn: ASCII file specifying language model in printable format.
\KVSLMOD: Directory of language models; created by INSTALL.

1. A downloaded phrase book uses enroll.kam and .kpm unless you enroll on an application vocabulary for more accuracy or make a new phonetic model (see Chapter 13). If you do either of the latter the phrase book will have 3 more files: {pbook}.kam, {pbook}.kpm and{pbook}.kpl.

KVMAKELM: HOW TO MAKE A PHRASE BOOK STARTING WITH A SPREADSHEET PROGRAM

Before the KVS can use an ASCII language model for recognition, the language model must be processed by KVMAKELM. KVMAKELM compresses the language model into KVS format and writes it into the \KVSLMOD directory.

You must run KVMAKELM after creating an ASCII format language model *and* after changing it.

To use KVMAKELM, type a command such as the following at C>:

KVMAKELM wordkit.prn [RET]

where wordkit.prn is the name of the .prn file you made with the spreadsheet program or a text editor. Type a space after KVMAKELM and after each command line modifier (see below).

KVMAKELM will look for the input .prn file in the current directory unless you type a full pathname such as \KVSUSERS\bill\newdates.prn. Then it will look for the .prn file in that directory.

KVMAKELM produces two files -- a .klm (Kurzweil language model) and a .kmp (Kurzweil make phrase for training) in the \KVSLMOD directory. These files are later copied into a phrase book for recognition.

Several single-letter modifiers on the command line change the way KVMAKELM works. All are optional except the name of the input .prn file. The full form of the command is

KVMAKELM [-c] [-l language model] [-r resolution] [-t tokens_per_phrase] [-v] input .prn file [RET]

The following modifiers come before the input file name.

[-c] designates a comma-separated list (ASCII format). The input file is a translation table where the columns are separated only by commas instead of the ASCII .prn export format shown in Figure 7.1. This form of input file is produced either by typing the language model into the computer with a text editor or by running KVLISTLM on a KVS format language model.

[-l language model] designates the name of the output language model. The name defaults to the name of the input .prn file if this argument is omitted.

[-r resolution] designates the resolution of each speech sample taken during training. The higher the resolution, the more exact is the KVS's internal image of the speech sound; higher resolution can improve recognition accuracy but takes more memory. The allowable

range is a number from 1 to 7; the default resolution is 4. A higher number gives a higher resolution but may result in slower recognition. (See also Chapter 12, [R] Resolution.)

[-t tokens_per_phrase] designates the number of times you say each phrase during training. An argument of -t 5, for example, specifies 5 tokens per phrase and is written this way:

c>KVMAKELM -t,5 DATES [RET]

Type a space after -t, -l, or -r and before the value. If your vocabulary is large, set [-t] to a small number, and use the <train> command to add extra tokens for confusable words only. The KVS sets [-t] to 3 by default. The legal range is from 1 to 100.

[-v] varies the order in which phrases are presented to the user for training. Varying the order helps the speaker supply a variety of inflections and pronunciations for the token set, and variety in a token set makes it more likely to contain good matches in recognition. No numerical value follows [-v]. It's either present or absent to tell KVMAKELM how to list the words for training.

If this argument is omitted, the KVS prompts for phrases in the same order for each repetition of the tokens in the vocabulary. If it is included, the order will vary. The default order of presentation is roughly the order in which phrases are listed in the PHRASE column of the language model. The more tokens of a phrase a user speaks, the more useful a varied order can be.

KVLISTLM: HOW TO PRINT A LANGUAGE MODEL

You can print a language model by running the Kurzweil AI utility KVLISTLM. The printed output can help users learn and remember the phrases in the vocabulary. KVLISTLM displays output on the screen or writes it to a print file if you redirect the output.

KVLISTLM lists phrases and translations in the language model by subset. A sample of the output is shown in Figure 9.1. A summary of the total number of phrase-translation pairs (excluding star phrases), subsets, and tokens specified by the language model appears at the head of the listing, followed by a two-column table for each recognition subset. The first column contains a numbered list of spoken phrases; the second column contains a translation string for each phrase. Star phrases are not counted in the summary since they are not trainable.

The command KVLISTLM WORDKIT [RET] produces this output:

```
    The following is a language model listing of "a:\kvslmod\wordkit:"
 -- generated from the recognition language model "wordkit";
 -- and compiled using kvmakelm Version 2.0
 -- which contains 2 recognition subsets.
 -- There are 27 trainable phrases
    to be represented by 81 tokens (at resolution 4).
Phrases and translation strings in the "COMMAND" subset:

1: *UNKNOWN              <BEEP><WINDOW That word is not in the current
                         vocabulary>
2: *ILLEGAL              <BEEP><WINDOW That word is not in the current
                         subset>
3: *ENTER                <WINDOW Listening for commands>
4: #DELETE LAST WORD     \b
5: #TRY AGAIN            \b
   .
   .
Phrases and translation strings in the "DIGIT subset":

1: *UNKNOWN     <BEEP><WINDOW That word is not in the current vocabulary>
2: *ILLEGAL     <BEEP><WINDOW That word is not in the current subset>
3: *ENTER       <WINDOW Listening for numbers><PARAM Wordgap 3>
4: *EXIT        <PARAM Wordgap 5>
5: #DELETE LAST WORD     \b
   .
   .
```

Figure 9.1 Sample Language Model Printout

Several command line switches modify the action of KVLISTLM. The full form of the command is:

KVLISTLM [?] [-s subsetname] [-t] [-p] {language model} [>printfile] [RET]

where:

[?] displays a usage note: the arguments to KVLISTLM.

[-t] lists phrases only, in training order; translations don't appear in the output.

[-p] lists phrases only, in phrase map order: translations don't appear in the output.

[-s subsetname] lists phrases and translations for one subset only. If "subsetname" is *phrase*, the output is a numbered list of all the

phrases in the vocabulary, numbered the way <edit> numbers phrases (including star phrases).

[-c] specifies that output is a comma-separated list (see Figure 9.2, below).

[-a] lists phrases alphabetically.

{language model} signifies the base name of the .klm and .kmp files to be listed. KVLISTLM will look for these files in \KVSLMOD, unless you supply the full pathname to your phrase book files, for example \KVSUSERS\frank\dates.

[>printfile] redirects output of KVLISTLM to the named print file. The named file may then be printed, typed, or edited with MS-DOS commands. If the argument is omitted, output is to the screen.

All switches and arguments are optional except {language model}, the name of the input file.

Examples (** are comments):

```
KVLISTLM wordkit          **lists the whole WORDKIT language model
KVLISTLM wordkit >wd.dat  **writes the listing to a print file for later
                            printing
KVLISTLM -s digits wordkit **lists only the digits subset
KVLISTLM -p -a wordkit     **lists all phrases in WORDKIT alphabetically
```

KVLISTLM: HOW TO RETURN A LANGUAGE MODEL TO ASCII FORMAT FOR EDITING WITH A SPREADSHEET

The KVLISTLM command line modifier [-c] produces an ASCII language model from your phrase book. This feature lets you take a language model you have created or changed with <edit> and further edit it with a spreadsheet program.

KVLISTLM produces the language model in the comma-separated list format shown in Figure 9.2. This is equivalent to 1-2-3 file import format and so can be imported into a spreadsheet. [n]umbers, not [t]ext, must be specified for the format. For example:

```
[in 123]   \[f]ile [i]mport [n]umbers
```

If you specify [t]ext, all fields will be copied into column A of the spreadsheet. If you specify [n]umbers, each field will be copied to a separate column.

Also, KVMAKELM will make a language model from an input file in this format. Notice that each phrase of the language model starts a line of the file. Each phrase is enclosed in quotes, and each translation is enclosed in quotes. A comma separates the quoted items. Missing

entries are indicated by two sets of quote marks surrounded by commas. Line continuation is shown here by a double backslash (not part of the import.prn format).

```
"PHRASE","COMMAND","DIGIT"

"*UNKNOWN", "<BEEP><WINDOW That word is not in the current\\
vocabulary>", "<beep>WINDOW That word is not in the current vocabulary>"

"*ILLEGAL","<beep><WINDOW That word is not in the current\\
subset>","<beep>"<window That word is not in the current subset>"

"*ENTER","<WINDOW Listening for commands>","<WINDOW Listening for\\
numbers><PARAM Wordgap 3>"

"*EXIT","","<PARAM Wordgap 5>"

"#DELETE LAST WORD","\b","\b"
```

Figure 9.2 Import.prn File Format for 1-2-3 Import Files

To produce the import.prn file, type, for example,

```
KVLISTLM \KVSUSERS\bill\wordkit -c >wordkit.prn
```

where:

\KVSUSERS\bill\wordkit signifies the full datapath to the phrase book.

-c tells KVLISTLM to produce the import.prn file format shown in Figure 9.2.

>wordkit.prn redirects the output to a printable file named wordkit.prn.

KVMAKEPB: HOW TO PUT AN UNTRAINED PHRASE BOOK IN THE END-USER'S DIRECTORY

When you or a user types <train>, a phrase book will be assembled in the KVS if none exists. However, you may wish to assemble the phrase book before the user trains it.

KVMAKEPB copies the language model files you specify with the user's enrollment voice profile to make a new phrase book ready for training. The resulting set of files is stored on the fixed disk under \KVSUSERS\{user's name}.

IMPORTANT

You must run KVMAKEPB if you are going to use a special voice profile (not ENROLL). <train> assumes the ENROLL voice profile.

To run KVMAKEPB, type, at the C> prompt,

KVMAKEPB [RET]

The KVS will prompt for some names. Here are the names to supply to make a phrase book for NEWDATES, for example

KVS User Name (default): {user's name}
New phrase book name ( ): NEWDATES [RET]
Language model name (NEWDATES): [RET]
Voice Profile name (ENROLL): [RET]

Notice that when a name is supplied in parentheses (ENROLL), you need only type a [RET] to accept it.

When the last prompt is answered, file copying begins. KVMAKEPB displays messages as it copies files over into the named user's subdirectory. You can ignore the messages until you see this one:

"YOUR NEW PHRASE BOOK IS READY"

The phrase book created in \KVSUSERS\{user's name} contains the language model and voice profile but is still missing the token file. It can't be used for recognition until a user trains it. The process of training adds a file of tokens to the phrase book, and links tokens from training with phrases from the language model. Training is described in Chapter 10.

KVMODPB: HOW TO USE EXISTING TOKENS WITH A NEW LANGUAGE MODEL

When you change a language model (add phrases or change their translations) using KVMAKELM after you have trained it, you must run KVMODPB to reassemble the phrase book in your directory. A flow chart of this process is shown in Figure 12.2.

KVMODPB makes a modified phrase book from an existing set of training tokens (.kvc file) and a new language model. All tokens for phrases that appear in the original phrase book are transferred to the new phrase book. However, new phrases in the changed version of the language model won't have tokens and must be trained. <train> knows which phrases are new and prompts you to say them.

KVMODPB runs on your host PC. The KVS can be on or off. The full form of the command is

KVMODPB{your name} {pbook name} [language model name] [RET]

where:

{user} designates the owner of this phrase book,

{pbook name} designates the phrase book to be modified,

[language model] designates the revised .klm file, if different from {pbook name} (optional).

If the phrase book to be modified is loaded in the KVS, save it and sign off as shown below before running KVMODPB (in our example, the user is Bill and the phrase book is NEWDATES):

```
                    <bye>
                    [Y] (yes, save changes to disk)
[at C>:]            KVMODPB
                    <restart>
                    <user bill>
                    <train newdates>
                    [S] (save phrase book)
```

Notice that after using KVMODPB, a user must sign off and restart the KVS before using the modified phrase book. This lets the KVS differentiate between old and new versions of the phrase book having the same name.

After training your vocabulary, you'll be able to use the KVS for voice input to your application. Training is described in the next chapter.

CHAPTER 10
TRAIN A PHRASE BOOK

CONTENTS:

- Starting Training with an <edit> Language Model
- Starting Training with an ASCII Language Model
- Recording Tokens
- Skipping Tokens During Recording
- Pausing and Resuming Recording
- Saving Training
- Adding, Replacing, and Deleting Tokens

This chapter explains how to train the KVS to recognize the phrases in your application vocabulary. Training creates a set of tokens, reference patterns of your voice saying each of the vocabulary phrases. At the end of training you save the tokens (.kvc file) to the PC's disk together with the other files in the application phrase book.

When you begin to use the KVS to recognize your speech, you load the tokens, language model and a voice profile into the KVS. The KVS recognizes what you say by matching tokens with your spoken input and sending the translations for the best matching tokens to your PC.

STARTING TRAINING WITH AN <edit> LANGUAGE MODEL

To move from <edit> into training, press [CTRL-Z] to return to the main <edit> menu. A choice of [RET] from the main menu gives the train function. You don't have to supply your user name and phrase book name to train; <edit> passes them along. Next you'll record tokens. If you have a problem starting training, read "If <train> Didn't Work" at the end of this chapter. Otherwise skip to "Recording Tokens," below.

STARTING TRAINING WITH AN ASCII LANGUAGE MODEL

Start up an application, sign on if necessary, and give the <train> command with the same output language model name you gave to KVMAKELM (in our example, the name is NEWDATES). Type at the C> prompt

```
KVTEST [RET]                    (optional)
<user {your name}>
<train newdates>
```

If the language model to be trained is already loaded in the KVS, just type a space and a period (.) after "train" instead of the phrase book name. This tells the KVS to use the currently loaded phrase book.

If you are an end-user working with a vocabulary developed for you, instead of typing <train> you can type, at C>,

```
KVTRAIN [RET]
```

to trigger all the events of training. KVTRAIN prompts for your phrase book name and user name.

After you initiate training by one of these methods, the KVS will tell you to stand by while it loads the language model into the KVS along with your voice profile (if it isn't already loaded), creating an empty phrase book for you to fill with tokens. Then you'll see the <train> menu. If you have a problem, please read "If <train> Didn't Work"

below. Otherwise, go on to the next section, "Recording Tokens."

RECORDING TOKENS

Recording tokens is the same process you went through to enroll. <train> prompts you to say phrases from your application language model. You'll say each phrase as prompted, and when the screen says "Done recording all tokens," you'll save your phrase book to the PC's disk. With the <train> menu in the window, press [A] to select Add tokens from the menu:

```
Train menu                                                    mic OFF
[A] Add tokens          [R] Replace tokens      [D] Delete tokens
[C] Count tokens        [S] Save tokens         [RET] Recognize
```

Choosing [A] gives the following window:

```
ADD TOKENS:                                                   mic OFF
Press [RET] To RECORD phrases or type spelling of phrase to Add
Add>
```

[RET] adds tokens for the entire vocabulary, so press [RET] to start recording.

The next window prompts you for a microphone check. Do the microphone check when you begin recording; do not press [ESC] or you may record bad tokens. After you say the test sentence, the KVS displays the message: "All done. Thank you." and switches immediately to recording tokens. You will be prompted to say the vocabulary phrases, one at a time:

```
1                                   [B] go Back             mic ON
2 SAY:          where am I          [P] Pause
3               enter commands      [S] Skip it             [esc] menu
```

Notice several differences from the ENROLL record window: you can decide to skip the next phrase -- not train it -- by pressing [S]. You can stop at any time and try recognition by pressing [ESC] and then [RET].

You will also see that the phrases to say come up much faster than during enrollment. The KVS is learning and processing as fast as you can speak.

Remember that training is a data collection process. You should take care with the process of recording:

1. If the screen skips ahead before you say the phrase, press [B] to go back and say the phrase again.

2. When the KVS prompts you to say a phrase, speak clearly.

3. Speak when text is displayed in the window, not before.

4. Don't speak too quietly or too loudly.

5. Pronounce phrases naturally, as you would in conversation.

6. If you make a mistake or cough or the phone rings, press [B] to go back and say the phrase again.

7. If you take off your headset or change the microphone position, press [P] to pause and [RET] to redo the microphone check.

SKIPPING TOKENS DURING RECORDING

The recording procedure is just like enrollment, except that you can skip a phrase [S] instead of saying it. Skip only if all of the following are true:

- Token memory is nearly full. When the main token memory is 100% full, the KVS recognition response time can slow slightly (see "Obtaining a Status Report on Training").

- You already have at least two tokens for the phrase.

- The phrase does not rhyme with or sound similar to any other vocabulary phrases.

If you choose [P] during recording, the window shows how close you are to finishing training, and how full token memory is:

```
Pause on token 80 of 794          10 percent done          mic OFF
                  Press [RET] to resume recording
Memory use: main 2%    reserve 0%    Average word length 0.59 seconds
```

PAUSING AND RESUMING RECORDING

Press [P] at any time during recording to take a break. When you are ready to resume, press [RET]. When you resume after a pause you will be asked to check the microphone again. Repeat the microphone check if you have taken off your headset or changed the position of the microphone. Otherwise, bypass it by selecting [ESC] from the microphone check menu.

Obtaining a Status Report on Training

Use [P] during recording to find out how much memory you have left for addition of tokens. The pause screen reports the percent of main memory used, the percent of reserve memory used, and the average word length in seconds.

Main token memory holds the number of tokens the KVS can scan in real time when it's matching speech against stored tokens. This is about two-third of capacity. The KVS also has a reserve token memory for the remaining third.

The KVS can have about 1,920 5/8-second tokens stored and give you recognition instantly. However, as average word length increases, more memory is used for each token. At an average word length of 1.2 seconds, for example, you have room for half as many tokens in main memory as you do with 5/8-second tokens.

As memory fills beyond this point, using the reserve memory slows recognition slightly.

Using main memory only, you will get recognition .27 second after you finish saying a word. That's about the length of the pause required (.25 second) for the KVS to know you have completed a word. If you use all of reserve memory as well, response time will go up to .42 second on 1/2-second tokens.

SAVING TRAINING

You know you're finished recording when the <train> menu returns with a message such as "DONE recording all 792 tokens."

Save your tokens to disk by choosing [S] from the train menu. If you forget, the KVS will remind you to save tokens when you type the <bye> command or try to load a different phrase book for training or recognition.

REMEMBER TO SAVE TOKENS

Your training is not permanent until the tokens are saved to disk.

While you are working back and forth among the <train>, <recognize>, and <edit> commands, tokens need not be saved each time you exit a command. The KVS will remind you to save your phrase book to disk when you give the <bye> command and also when you try to

load another phrase book. <recognize>, <edit>, and <train> all save the same information.

Your next step is to test recognition with the tokens you've made,

ADDING, REPLACING, AND DELETING TOKENS

First, a little detective work. How well are your tokens performing?

Start up KVTEST (KVTEST [RET]). Put the KVS in Ordered_list output mode (<output o t>), start up recognition (<recognize .>) and say the phrases in your vocabulary. Instead of seeing one phrase on the screen, you'll see a list of up to 5 alternatives for that phrase, each preceded by a number. The number is a recognition score. Note the score of the top phrase and any phrases scoring close to it. If the top phrase is not the one you said, and the next alternative is close in score, the KVS is confusing them. If the top phrase is correct but the next alternatives are very close in score, the tokens are potentially confusable.

KVTEST reports scores that are higher for better recognitions and lower (falling toward zero) for more uncertain recognitions. The higher the score, the closer the spoken phrase is to the tokens stored in your phrase book. Recognition scores nearing 1,000 are very good. Remember, though, that it's the consistency of recognition, not simply one score, that determines how well the KVS performs as a recognizer. If one phrase never gets a score above 400 but is never confused with another phrase, that phrase will always be recognized correctly.

You can add, delete, or replace tokens to improve recognition depending on your recognition scores. Table 10.1 lists the actions you can take:

Table 10.1 Improving Recognition

Problem	Solution
Top choice scores high -- 600 or more -- but phrase in second position is close.	Add 2-3 tokens to both.
Top choice scores low -- 400 or less.	Replace tokens.
Scores vary from one recognition to the next.	Add and/or replace tokens.

Table 10.1 (cont.) Improving Recognition

Problem	Solution
Odd matches are proposed for phrase "x."	Replace tokens of "x."
Intruding noises trigger false recognitions for phrase "x."	Replace tokens of "x."
When you say phrase "x," KVS responds with phrase "y" and they sound alike.	Add tokens of "x" or use a synonym. (Add the synonym, if the vocabulary needs one, or put the phrases in different subsets.)
Phrase "x" is seldom recognized.	Replace tokens.
Phrase is recognized only when said a certain way.	Say it that way or add tokens.

The <train> command lets you add, replace, and delete tokens. To return to training from recognition, just type

<train .>

When you see the <train> menu, you can add, delete, or replace tokens for a phrase.

Adding Tokens

Press [A] to add tokens, and type one of the following commands.

To record 2 more tokens of a {phrase},

Add>{phrase} [RET]

To record [n] more tokens of a phrase (n = 1-9),

Add>*{n} {phrase} [RET]

Leave a space between the number of tokens {n} and {phrase}. If you type, for example,

Add>*5 halyard[RET]

the prompt to say "halyard" will be repeated 5 times. When adding tokens, keep the total number of tokens per phrase to six or fewer. More than a total of 6 tokens per phrase probably does not add useful information.

After you have said the phrase the specified number of times, the <train> menu returns. Press [RET] to return to recognition. Press [CTRL-Z] to exit KVTEST when you're finished with the training-testing cycle.

Replacing Tokens

A choice of [R] (replace) from the main <train> menu does a one-for-one replacement of all tokens stored for the phrase you type. You are prompted to say as many tokens of the phrase as you have stored in your phrase book. The new tokens you speak are stored in place of the old ones.

Deleting Tokens

A choice of [D] (delete) from the main <train> menu deletes all stored tokens of the phrase you type but does not delete the phrase from the language model. [D] reports the number of tokens deleted. If, after deleting tokens, you want to add tokens for that phrase, simply press [A].

Counting Tokens

To learn how many tokens your phrase book has stored for a given phrase, ask the KVS to count tokens [C] for that phrase. [C] asks you to type the phrase and then produces a report such as

n of m tokens trained for {phrase}

where "n" is the number of tokens stored and "m" the number the language model says should be trained.

Chapter 11 continues the discussion of recognition testing.

If <train> Didn't Work. . .

This section lists the error messages <train> gives and explains what to do about them.

1. "No voice profile available. No training allowed."

   The enroll.klm and enroll.kpm files are missing from your user's directory. You must record tokens for the enroll vocabulary before you can train another vocabulary. See Chapter 4, KVENROLL. See also Chapter 13, "Enroll on an Application Vocabulary." To use a vocabulary-specific voice profile, see Chapter 12, "[V] Voice Profile."

2. "Phrase book is incomplete."

   The named phrase book is missing one or more of these files: .kam, .kpm, or .kvc. If all three are missing, repeat enrollment from the beginning. See Chapter 4.

3. "Phrase book is missing."

   The language model files (.klm and .kmp) are not present in either \KVSLMOD or your user's directory. No training can be done. To supply these files, you must develop a vocabulary. Please see Chapters 6-8.

CHAPTER 11
TEST A PHRASE BOOK: MORE ON RECOGNITION

CONTENTS:

- What Performance Should you Expect of the KVS?
- If the Unexpected Happens. . .
- Find Problem Phrases: Ordered Output Mode
- Check Subset
- Adjust Rejection Threshold
- Delay KVS Message Transmission: <!Wait>

This chapter explains how to test your training tokens. It also covers common problems that show up when you try out a new vocabulary.

WHAT PERFORMANCE SHOULD YOU EXPECT OF THE KVS?

In most applications the KVS is at least 95% accurate as a recognizer. This means it understands 19 phrases out of 20 on the average for selected vocabularies. Accurate automatic speech recognition, like human communication, however, depends on several factors:

- Whether the spoken phrases sound alike to the recognizer
- How much background noise interferes with listening
- How clearly you are speaking
- How well you've prepared (trained) the KVS
- Whether one device talks to the other before the other is ready to listen (communications protocol)

These factors affect performance in different ways. To get an idea of how the KVS should respond, this chapter suggests a test, using the Newdates phrase book as an example. If you have trained that phrase book, try recognition on the phrases shown below, or run a similar test on a phrase book of your own. Start up recognition in the usual way:

To	Type
Sign on as a user	<user {your name}>
Start recognition of the vocabulary	<recognize newdates>
Wait for the vocabulary to download	KVS prompts: Downloading newdates Please stand by.....
Run the microphone check	[RET]
Run KVTEST, say "run kvtest" or	KVTEST [RET]

Then try this:

You say:	The screen types
"set dates"	Enter new date (mm-dd-yy):
"April two two eighty-six"	4-22-86
"microphone off"	Microphone is off
"January three eighty-two"	(no response)
"listen to me"	Listening for dates
"January three eighty-two"	1-3-82
"where am I"	subset months (or) subset command

IF THE UNEXPECTED HAPPENS. . .

If you got an unexpected response, read on. This section recommends actions for dealing with common problems. It also tells you where the recommended action is explained in the manual.

1. *System confuses phrases that sound alike.* You say one phrase; the system types another that sounds like the phrase you said.

   - Find all problem phrases. See "Find problem phrases," below.
   - Retrain both phrases: replace all tokens for both phrases, and retest. See"Replacing Tokens," Chapter 10, and "Find Problem Phrases," below.
   - Add synonyms. See Chapter 12, "The <edit> Functions" and "[A] Add phrases," and Chapter 8 "Choosing Spoken Phrases."

2. *System seems deaf.*

   - Check subset. See "Check Subset," below.
   - Check microphone position. See Chapter 3, "Microphone Placement."

- Reset rejection threshold. See "Adjust Rejection Threshold," below.
- Run a microphone check (if the *green* speaking-level indicator does not flash, raise the gain). See Chapter 4, "Testing One Two... ." See also Chapter 13, "How to Minimize Effects of Background Noise."

3. *Accuracy falls off suddenly, while you are speaking.*
   - Run microphone gain test. See Chapter 4, "Testing One Two..." See also Chapter 13, "How to Minimize Effects of Background Noise."

4. *System misses short phrases (six, up, button).*
   - Resay the phrase, making the vowel portion longer.
   - Retrain phrases. See "Replace Tokens," Chapter 10.

5. *Phrase appears on the screen when you're not speaking.*
   - Replace tokens for the phrases that appear on the screen. See "Replace Tokens," Chapter 10.

6. *A spoken command doesn't execute correctly.*
   - <edit> translation string for current subset.
   - Put a delay <!wait> in the translation string. See "<!wait>," Chapter 8, and Appendix B.
   - Does a rhyming word with a different translation exist? If so, see "Find Problem Phrases," below.

FIND PROBLEM PHRASES: ORDERED OUTPUT MODE

Problem phrases are phrases that either produce errors or sound similar to other phrases in the vocabulary and can therefore be called potential errors. Many of them will simply be rhyming words you see in the candidate sets when you run the KVS in ordered output mode.

This section tells you how to use ordered output mode to find problem phrases and what to do about the problem phrases you find.

Using Ordered Output Mode

The command

<output ordered_list typeahead>

changes the output mode for the recognizer. Instead of typing the single best-scoring token when you say a phrase, the KVS types a list of the top-scoring tokens and their scores. The list looks like this:

User says "wake." KVTEST types:

1010 wake

845 wait

577 grate

Another example:

User says "down." KVTEST types:

919 gown

845 down

552 guide

542 jump

KVTEST reports higher scores for better recognitions and lower scores (falling toward zero) for more uncertain matches. The higher the score, the closer the spoken phrase is to the tokens stored by the KVS. Recognition scores around or above 1,000 are very good. Scores lower than 500 are more indecisive. Tokens of similar sounding phrases with similar scores are potential problems.

To test the vocabulary, follow this procedure:

1. Start up KVTEST. Put the KVS in ordered output mode,and start recognition of the phrase book you are testing

   (a) Type KVTEST or say "run KVTEST."

   (b) Type <output ordered typeahead>.

(c) Type <param maxphrases 5>.

(d) Type <recognize newdates>.

2. Say each phrase in the vocabulary. (See Table 7.1 for the phrases in NEWDATES.)

3. Write down any pairs or triplets in a candidate list that sound alike and have similar scores (many of these will be rhyming words).

4. Do two things with the list of similar sounding phrases:

   (a) Replace tokens for the confusable phrases.

   In the examples above, replace tokens of "gown" and "down." (Replacing tokens is explained in Chapter 10.) You don't have to replace tokens for "wake" and "wait." Even though the phrases sound similar, the token scores are far enough apart to distinguish the phrases in recognition. The tokens you have trained are good.

   (b) If you are not the end-user of this vocabulary, change the language model to take more tokens of these phrases (see KVMAKELM, Chapter 9, and Chapter 12, <edit>, "Setup Menu," and "[T] Tokens Per Phrase").

5. Return the KVS to top choice mode. Type

   <output top_choice typeahead>

For more information about recognition modes, see Appendix B, "Angle Bracket Commands," <output>. For information about <param maxphrases>, see Appendix B, <parameter>.

The size of the candidate list in ordered output mode is controlled by two settings:

- The maximum size of the list is limited by the value of maxphrases. If it is set to <param maxphrases 1>, only the top choice appears. If it is set to <param maxphrases 5>, the list is limited to 5 candidates.

- The minimum list is controlled by the rejection threshold as explained below. Phrases scoring below the threshold are discarded. When no phrase scores above the threshold, KVTEST reports "No good match."

Use the other items in the candidate list to remind you of other phrases in the vocabulary to test.

CHECK SUBSET

The active subset may lack a translation for the phrase you say. If it does, the KVS will not respond when you speak.

Some language models will tell you when you say a phrase that is not in the active subset or in the vocabulary (see Table 8.1, *ILLEGAL and *UNKNOWN). If yours does this, it will display a window message rather than appear deaf.

If you get no response at all when you talk or you get a message such as

?? PHRASE IS ILLEGAL

or

?? PHRASE IS UNKNOWN

find out which subset is active by saying "where am I" or typing

<subset

The KVS will display the name of the subset, such as

The current subset is months

This information tells you if you are in a different application context (such as digit entry) from the one the KVS is listening for (such as text entry). If the subset is correct, skip to "Adjust Rejection Threshold." If the recognition subset is wrong, change to the right one by typing

<subset {name}>

where {name} is the name of the translation subset that you want.

When you give this command and resay the phrase, the KVS should no longer act deaf.

Some language models let you switch subsets by voice. If you are recognizing with the NEWDATES phrase book, for instance, you can say "subset command" to switch into the command subset, or "set date" to switch into the months subset. Both of these phrases translate to the <subset {name}> command.

To display all subset names in the phrase book, type

<subset ?>

ADJUST REJECTION THRESHOLD

The rejection threshold is a recognition score that limits the candidate list to tokens that score above the threshold. If background noise is high and the KVS acts deaf, lowering the rejection threshold may help. When you lower the threshold, more phrases qualify as recognition candidates.

The threshold can have a value between 0 and 9. To see the current threshold value, type

<thresh>

To make the KVS transmit more of what it hears, lower the threshold from 5 (the default) to 3 by typing the command, a space, and the number, for example:

<thresh 3>

Continue speaking.

IMPORTANT

Remember that lowering the threshold can result in another kind of error: mistaking background noise for a phrase. If this happens, raise the threshold again.

When no phrase scores above the threshold, the KVS returns whatever translation the language model supplies for *UNKNOWN. If only phrases outside the active subset score above the threshold, the KVS does not return the phrase but instead returns whatever translation the language model supplies for *ILLEGAL.

DELAY KVS MESSAGE TRANSMISSION WITH <!WAIT>

If a spoken command doen't execute correctly, the KVS could be sending recognition results to the application before the application is ready to receive them. To test for this, introduce a delay in the translation string for the problem phrase. For instance, suppose a user says "outside line" to get the attention of a modem and have it dial a number outside the company. Because there is a short delay in getting an outside line, the modem may be receiving the phone number before it's ready to dial. Here is the voice command without the delay:

"outside line" ATDT893-5151\r

You want the KVS to wait after getting the modem's attention, so you insert the delay after the attention request and before the phone number.

"outside line" ATDT<!wait 2000>893-5151\r

The <!wait> command takes an argument in milliseconds (thousandths of a second): 2000 milliseconds equals two seconds, 500 milliseconds equals .5 second.

CHAPTER 12
CHANGE A PHRASE BOOK

CONTENTS:

- Why Change Your Phrase Book?
- Changing the Written Part of Your Phrase Book
- <EDIT> Features
- <EDIT> Functions
- What You Have Learned

WHY CHANGE YOUR PHRASE BOOK?

After training a phrase book and trying to recognize with it, you may discover that a phrase or two are missing or that some of the phrases are hard to say. Erroneous translation strings may be sent to the host PC. To add or change phrases and change translations, you will need to make changes to the written part of your phrase book -- the language model.

<edit> is the best tool for changing a language model because you can so easily go from editing to training and then try out your changes in recognition. Anything you can do with a spreadsheet, you can also do with <edit>.

Also, you can run <edit> functions by voice if the language model you are editing contains the #edit subset (see "Add phrase" example, Chapter 8).

Recognition testing sometimes reveals that the spoken part of a phrase book -- your tokens -- also need changing. Pairs of phrases may confuse the KVS, or patterns you made in training may not perform as well as you expect. To increase the recognition accuracy of the KVS, you can use the <train> command to add, replace, or delete tokens. Use of <train> to change tokens is discussed in Chapter 10. This chapter describes the use of <edit> to change a language model.

CHANGING THE WRITTEN PART OF YOUR PHRASE BOOK

Table 12.1 shows how to improve recognition performance when recognition testing turns up any of the common problems listed below. All these improvements require changes to the language model.

TABLE 12.1 Improving Recognition Performance by Changing the Language Model

Problem	Solution
New tasks require additional vocabulary.	Add phrase.
The user needs additional feedback.	Provide more help commands and window messages.
Two phrases are frequently confused.	Change one of them to a more distinctive phrase, or add a synonym, or put each phrase in a separate subset.
A very short phrase is often missed.	Substitute a longer phrase.

Figure 12.1 shows the phrase book modification cycle using <edit>. With <edit>, you change your language model, train the changes, save the changes in your phrase book, try recognition, and you're done.

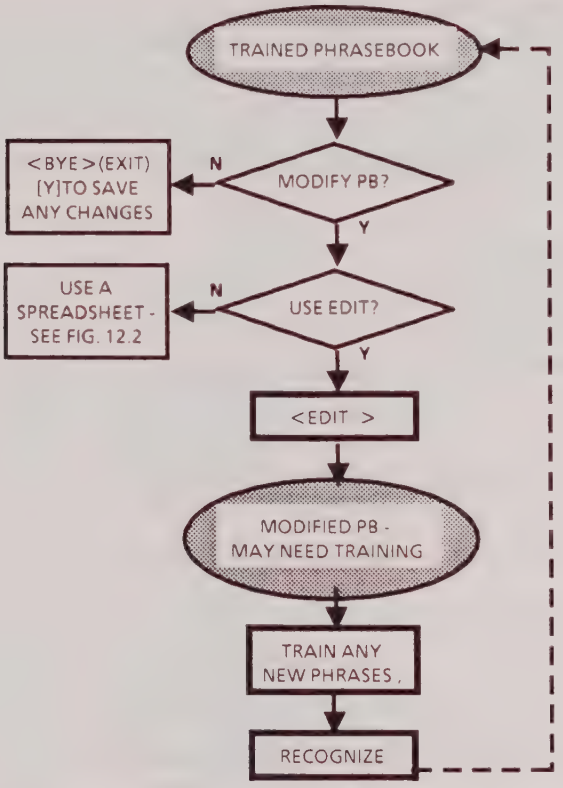

Figure 12.1 Phrase Book Modification CycleUsing <edit>

As shown in Figure 12.2, when you use a spreadsheet program to modify a language model, before you can train your changes and test them in recognition, you must

- return the language model to spreadsheet-input-file format (KVLISTLM);
- run the spreadsheet program on the input file (modify the language model);
- compress the spreadsheet output file into language model format (KVMAKELM);
- assemble the new language model with your phrase book (KVMODPB and <restart>).

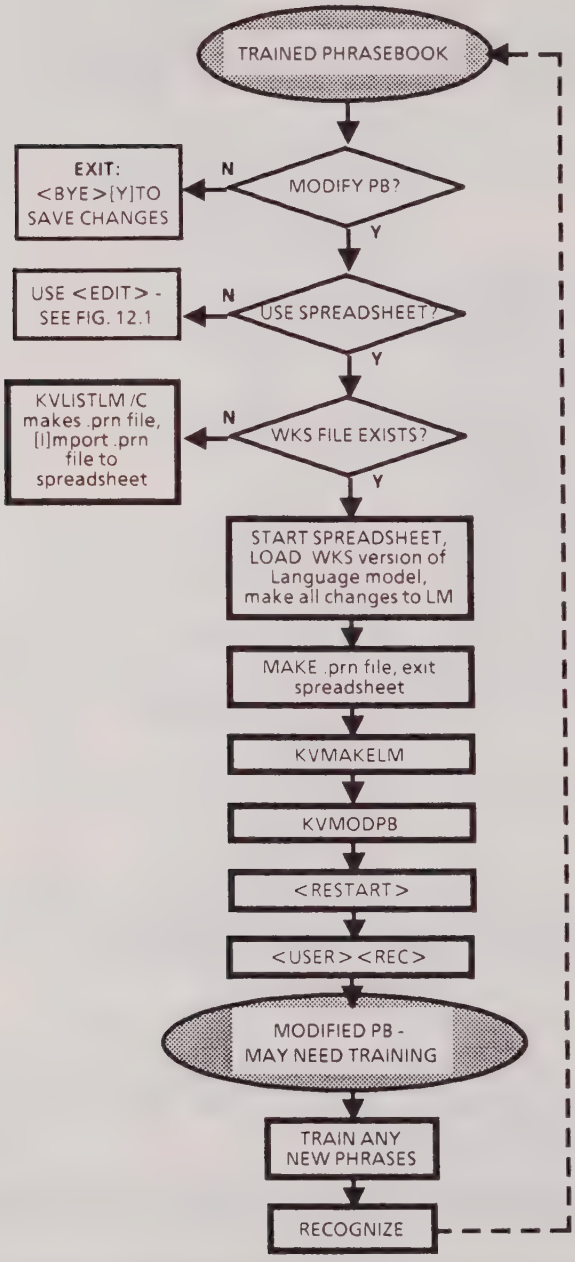

Figure 12.2 Phrase Book Modification Cycle Using a Spreadsheet

<edit> FEATURES

Chapter 7 showed you how to use <edit> to create and edit a language model. This chapter describes other features of the <edit> command. Functions are separately described in the next section and summarized in Tables 12.2 and 12.3 and Appendix B.

You can use <edit> to change the language model component of a phrase book, specify the voice profile component of a phrase book, set up a phrase book for recording tokens, and save or delete the work you do in an editing session. Anything you can do with a spreadsheet program and the KVS host utility programs you can also do with the <edit > command. The <edit> functions include the following:

1. Phrase, subset, and translation editing
2. Phrase book file management
   (a) Saving, deleting, and listing your phrase books
   (b) Merging language models from two phrase books
   (c) Merging a changed language model and a token set to make a modified phrase book
   (d) Publishing a language model to make it available for other users
   (e) Making a new phrase book from a published language model
3. Reporting of language model contents and size (information and vocabulary list)
4. Setup of phrase books for data collection
   (a) Including a vocabulary-specific voice profile with the phrase book
5. On-line help -- listing editing options

<edit> also has many features to make phrase and translation editing easier. These include the following:

- Shortcuts for typing in the language model (repeat and remember keys)
- Wild cards for finding phrases, translations, or subsets to edit
- Subset creation, deletion, and copying
- Addition/deletion of vocabulary on the fly
- Ability to work down one column (subset) or across a row (all subsets) as you type in your changes

Using the <edit> Command

To use <edit>, you'll need a connected KVS, but you won't necessarily speak to it. You can build your language model one day and train it and try it out the next. But you must sign on as a user, and you can't do that until you have enrolled.

To invoke <edit>, type the command, followed by a phrase book name (or a period for the presently loaded phrase book), for example:

```
<user ray>
<edit newdates>
```

If the phrase book exists, the KVS downloads it with these messages:

```
Loading phrase book newdates
Please stand by....
```

While the download is in progress, dots paint across the screen. When the download is complete, the main <edit> menu appears in the KVS window. Please skip to "Making a Selection from the Menu."

Making a New Phrase Book from a Published Language Model

If the named phrase book doesn't exist on the disk under \KVSUSERS\{your name}, <edit> searches the \KVSLMOD directory to see if it contains a public language model of that name. If it finds one, <edit> downloads the public language model with the message

```
Loading public language model newdates
Please stand by ...
```

followed by the <edit> menu.

Your new phrase book will contain all phrases and translations in the public language model, and the ENROLL voice profile. You can then use <edit> to add or change the existing phrases.

Making a New Phrase Book

If <edit> finds neither the phrase book in your directory nor a published language model under that name, it asks if you want to create one:

```
There is no phrase book {newdates}
Press   [Y] to create one
        [N] to exit
```

A reply of [N] exits the <edit> command. A reply of [Y] creates an empty phrase book containing only the ENROLL voice profile, which <edit> downloads with these messages:

```
Creating Phrase Book {newdates}
Please stand by ...
```

You can then use <edit> to create the phrases and translations in the language model, as demonstrated in Chapter 7.

Remember that a phrase book that <edit> creates for you exists only in the KVS and not in your user directory on disk. Therefore, be sure to explicitly save the phrase book to disk before you download another phrase book,or sign off.

Selecting an Item from the Menu

After downloading a phrase book, <edit> gives the first of a two-page main menu of functions.

```
EDIT MENU: page 1 of 2    Phrase Book:Newdates  [N] Next menu page     Mic OFF
[A] Add phrases           [E] Edit phrases      [V] Vocabulary list menu
[D] Delete phrases        [F] Files             [RET] Training
```

Notice that the name of the function on the menu is preceded by the name of the key [in square brackets] that selects the function.

Menu selections are made with a single key press. It is not necessary to type [RET] after making a selection. Selection of an item clears the menu and brings up a new window. The new window can contain another menu or the first screen in the chosen function. You can select a page 2 main menu function from page 1 or a page 1 main menu function from page 2. You do not have to be looking at the page containing the function.

Within a function, you will be prompted to supply the name of the phrase, translation string, subset, or phrase book you want to edit or query. The <edit> cursor (an underbar_) indicates the field to be filled in and the character position within the field where new typing will appear.

To return to the main menu from a function, type. CTRL-Z (or press [ESC] a number of times). [CTRL-Z] from within a function returns you immediately to the main menu.

Table 12.2 is an index to the <edit> functions, which are listed by menu.

TABLE 12.2 -- Index to <edit> functions

MENU AND ITEM		Described on Page
MAIN MENU ITEMS		
[A]	Add phrases	12-14
[D]	Delete phrases	12-15
[E]	Edit phrases	12-16
[M]	Make new subsets	12-17
[R]	Remove subsets	12-18
[X]	Change subset name	12-19
[C]	Copy a subset	12-20
[F]	Files	12-21
[I]	Information	12-26
[V]	Vocabulary list	12-26
[RET]	Training/Recognition/Exit	12-12
[N]	Next menu page	12-7
[S]	Setup menu	12-30
FILES MENU		
[L]	List phrase books	12-21
[S]	Save phrase book	12-22
[D]	Delete phrase book	12-23
[P]	Publish phrase book	12-24
[M]	Merge phrase book	12-25
[RET]	Escape to main menu	12-7
VOCABULARY LIST MENU ITEMS		
[P]	List phrases	12-27
[S]	List subsets	12-28
[T]	List phrases, subsets, and translations	12-29
[RET]	Escape to main menu	12-7
SETUP MENU		
[R]	Resolution	12-30
[T]	Tokens per phrase	12-31
[C]	Condense language model	12-32
[V]	Voice profile	12-33
[RET]	Escape to main menu	12-7

HELP: List Editing Options

<edit> provides editing keys and "remember" and "repeat" keys for entering phrases and translations. To see them, type [CTRL-O] [^O] (for options) inside a function. [CTRL-O] gives a series of screens listing cursor movement keys, delete keys, wild card and phrase completion keys, and more. Page through the options by typing [RET], or escape back to the function by typing [ESC]. Table 12.3 lists all the editing options.

TABLE 12.3 -- Editing Options [^O]

Key	Function
	To Step Through the <edit> Menu
[CTRL-Z]	Zaps back to main menu.
[ESC]	Returns to editing function.
[RET]	Displays next options page.
[N]	Next menu page (from main menu).
	Cursor Movement
[CTRL-B]	Moves cursor back one character.
[CTRL-F]	Moves cursor forward one character.
[CTRL-E]	Moves cursor to end of string.
[CTRL-A]	Moves cursor to beginning of string.
[CTRL-S]	Moves cursor to subset field. In [A] (add) or [E] (edit), [CTRL-S] works two ways: (1) An answer of [CTRL-S] to "which phrase?" or "edit phrase:n" jumps to the subset field so you can choose a subset to do all your adds or edits in. You type a subset name. After that, the function returns to the original question so you can specify a phrase within the subset. (2) An answer of [CTRL-S] to "edit translation" jumps to the subset field so you can change subsets before typing the translation.
[CTRL-N]	Next. Horizontal scroll to the right: shows next part of a translation string longer than 80 characters. A

TABLE 12.3 (continued) Editing Options [^O]

Key	Function
	marker of the form n/m appears at the lower right, where "m" signifies the length of the string in bytes and "n"signifies the cursor position in the string. For example, 50/180 means that the cursor is on the 50th character of a translation string that is 180 characters long.
[CTRL-L]	Last. Horizontal scroll to the left: shows previous part of a translation string longer than 80 characters. A marker in the lower right tells which portion of the translation string you're looking at (see [CTRL-N], above).
	Delete
[BCKSP]	Deletes one character to the left of the cursor.
[CTRL-C]	Clears the string being edited.
[CTRL-K]	Kills string from cursor position to end.
[CTRL-U]	Kills string from beginning to cursor.
[CTRL-D]	Kills one character to the right of the cursor.
	Searching: Wild Card and Phrase Completion
[CTRL-W]	Wild card. Used in search queries; matches any other character, including the null character.
?	Shows available choices (phrase books, language models, or subsets). Also automatically completes phrases, subset names, language model names, and phrase book names.
	Repeat and Remember
[TAB]	Inserts the current phrase, the current translation, or the current subset in the window. For example, an answer of [TAB] to "type translation" echoes the current phrase in the translation field. Pressing [TAB] again repeats the insertion.
[CTRL-P]	Puts the translation string in storage. Used with long translations or many subsets to insert the same string in every subset. Once a string is stored, you don't have to retype it to enter it; you just change subsets and type [CTRL-G] (see below). .

Table 12.3 (continued) -- Editing Options [^O]

Key	Function
	You must type [CTRL-P] when the string is on the screen and before entering it with [RET]
[CTRL-G]	Gets translation string from storage and inserts it in the translation field.
[CTRL-R]	Restores field contents to their original state, before editing.
	Microphone ON/OFF
[CTRL-V]	Toggles microphone ON/OFF. If the microphone is ON in editing, pressing [CTRL-V] turns it off. If the microphone is OFF [CTRL-V] turns it on.
	Learn Mode
[CTRL-T]	Teaches the KVS the translation of a phrase. In [A] (add) or [E] (edit) when prompted to supply the translation of a phrase, if you type [CTRL-T], the KVS window disappears, signifying that the program you are typing to now is your application. <edit> suspends its own operation and monitors the conversation with your application. It stores everything you type until you again type [CTRL-T]. When you type [CTRL-T], the KVS window reappears with the stored keystrokes in the translation field. Any nonprinting keys you entered appear as octal strings, for example \015 for [RET]. To accept the translation as is, type [RET]; to change it, use [^O] and then type [RET] to enter it. Learn mode can be turned on before you enter a translation string or after you've entered part of it.
	Entering Nonprinting Characters
quotechar	Next character is interpreted literally rather than as an ASCII code. Type a quote character before nonprinting characters like [ESC] or [RET] and before the quote character itself (') if you want to use them in a translation string. Quote converts the next character you type to an octal code and displays the code in the <edit> window.

A Word About Wild Cards

The wild card character finds fields or names whose contents you don't know or can't remember. Table 12.4 shows how to use wild cards to search for names of phrases, translation strings, subset names, or phrase book names you want to edit. You can specify one or more parts of a name, and <edit> will find all names that have those parts you specify. You can also type just a wild card for a name and <edit> will display all phrases or translations, subsets, or phrase books. All searches are case-insensitive. In the table, ^W signifies the wild card character, [CTRL-W].

Table 12.4 -- USE OF WILD CARDS TO SEARCH A LANGUAGE MODEL

Where to put [^W] in a search string	Examples of phrases found by <edit>
[^W]	All names (phrases, subsets, translations, phrase books)
Mode[^W]	Modest Mode change
[^W]mode	Command mode Date Mode
[^W]mode[^W]	KVS Model One
[^W]ee[^W]	Milwaukee Queens
[^W]key[^W]window	The key is on the window

Leaving <edit>

To move from <edit> into training or recognition, return to the main <edit> menu via [CTRL-Z] and then press [RET]. When your phrase book contains added or changed phrases, a choice of [RET] from the main menu gives the <train> function. Otherwise [RET] takes you into recognition. The main menu lists which function [RET] will give.

The next section describes <edit> functions individually. Functions are grouped by submenu:

1. Phrase, subset and translation editing
2. Phrase book file management
3. Reporting language model size and contents
4. Set-up of phrase books for data collection

THE <edit> FUNCTIONS

Phrase, Subset, and Translation Editing

The seven functions described in this section are used to add, delete, or change phrases, translation strings, and whole subsets. To work on your language model column by column, choose the subset functions. To work row by row, choose the phrase functions.

Edit Menu: page 1 of 2	Phrase Book: Dates	[N] Next menu page mic OFF
[A] Add phrases	[E] Edit phrases	[V] Vocabulary list menu
[D] Delete phrases	[F] Files	[RET] Training

NAME: **[A] Add phrases**

FUNCTION: Adds a phrase and all its translations to your language model.

PURPOSE: To add vocabulary on the fly

ARGUMENTS: phrase, translation string, subset name

COMMENTS: Pressing [A] starts entry of a row of your language model. Type the contents of the PHRASE column -- the phrase followed by [RET]. Next type the translation of that phrase in the first subset, followed by [RET]. If the phrase is not recognized in that subset, enter a null translation by typing only [RET].

If your language model has additional subsets, the function will let you add a translation string or a null translation to each subset column. To skip subsets and go to the next phrase, type [CTRL-S] and [RET].

All of the editing options can be used when you are supplying phrases, translations and subset names. See Table 12.3 for the use of [CTRL-S], to select a column for adding translations to. See also the use of the remember [CTRL-P] and repeat keys, [CTRL-G] and [TAB], for a shortcut in typing in translations.

See Chapter 8 and Appendix A for valid characters in translation strings.

After adding the last row of phrase-translation pairs, if you press [RET] twice at the "type phrase" prompt, you'll pop up to the main <edit> menu and then to training, where you will be prompted to say the phrases you just added. However, if you're using <edit> to make a phrase book with a vocabulary-specific voice profile instead of establishing tokens with the <train> command, escape from training and run KVENROLL (see Ch. 4).

NAME: **[D] Delete a phrase**

FUNCTION: Removes a phrase and all its translations from a phrase book.

PURPOSE: To recover space for additional vocabulary.

ARGUMENTS: phrase name

COMMENTS: This function removes all translations and all tokens of each phrase it deletes.

To remove a single translation from the language model -- make the phrase unknown in one subset and recognizable in the others -- use [E] (edit phrase) and delete the entire translation string.

Specify the phrase(s) to be deleted by typing the name of a phrase, part of a name combined with a wild card, or just a wild card. Entry of a wild card alone is equivalent to a request to delete all phrases in the language model.

If you use a wild card, DELETE reports the results of the wild card search before going ahead with a deletion. For instance, it will tell you how many phrases it found that matched your specification and ask if you want to see them. You can press [L] to see the wild card matches before deletion or [D] to delete all the matches.

Phrases lacking translations in any subset are also removed from the phrase book when condensing is done during [S] (save) or [P] (publish) phrase book. (See "Setup of Phrase Books for Data Collection, [C]," below.)

NAME: **[E] Edit phrases**

FUNCTION: Changes existing phrase spellings and/or translations.

PURPOSE: To edit vocabulary on the fly: change the spelling of an existing phrase and/or its translations. To make a phrase unrecognizable in one or more subsets by deleting its translation from one or more subsets.

ARGUMENTS: phrase name, translation strings

COMMENTS: Pressing [E] starts changes to a row of your language model. Select the row you want to edit by typing a phrase (or a wild card or a part of a phrase and a wild card) followed by [RET]. You'll be asked if you want to change the spelling of that phrase. If you do, correct the spelling and enter the correction by pressing [RET]. Next you'll be shown the translation of that phrase. Correct the translation if desired and press [RET] to enter the change. The function then cycles through the subsets, allowing you to edit translations across the row, skipping the subsets in which the phrase is not recognized.

Entry of [RET] after a translation string gives the next subset and its translation. Entry of [ESC] after [RET] escapes to the main menu.

All of the editing options can be used when you are supplying phrases, translations, and subset names. See Table 12.3 for the use of [CTRL-S] to select a subset for adding translations to. See also the use of the remember [CTRL-P] and repeat keys, [CTRL-G] and [TAB], for shortcuts in typing translations.

See Chapter 8 for valid characters in translation strings.

After editing the last phrase-translation pair, press [CTRL-Z] to return to the main <edit> menu, and then to training or recognition. Train will prompt you to say phrases you've changed.

NAME: **[M] Make new subsets**

FUNCTION: Creates additional translation columns (subsets).

PURPOSE: To expand a language model by adding one or more translation columns.

ARGUMENTS: subset name

COMMENTS: Pressing [M] prompts for the name of subset number n, where n is the first unused column in the language model. Type a subset name and press [RET] to enter the name you have typed. After that, you can continue making new subset columns or press [CTRL-Z] to return to the main menu. Subset columns are numbered sequentially from left to right. The first subset after the PHRASE column is numbered 1.

Subset names may be spelled with any printable ASCII characters. They may have 25 or fewer characters and have no embedded spaces. Two subset names are reserved for KVS use: #edit and #voice_command. (Use of these subsets is explained in Chapter 8.)

NAME: **[R] Remove a subset and all its translations**

FUNCTION: Removes one or more translation columns from a language model.

PURPOSE: To adapt a borrowed language model, correct errors, or recover space for additional translations.

ARGUMENTS: subset name

COMMENTS: Specify which subset to remove by typing either a subset name, part of a name combined with a wild card, or just a wild card. Entry of a wild card alone is equivalent to a request to delete all translation columns.

If you use a wild card, [R] tells you how many subsets match your specification and asks if you want to see their names. You can press [L] to list the matches before deletion,or [D] to delete all the matching subsets. This operation deletes all translations entered in the subset.

Removing a subset can leave some phrases without any translations in the language model. These phrases and their tokens are removed from the phrase book when condensing is done during [S] (save) or [P] (publish) phrase book. (See "Setup of Phrase Books for Data Collection, [C]," below.)

NAME: **[X] Change subset name**

FUNCTION: Changes the spelling of a subset name.

PURPOSE: To edit a subset name.

ARGUMENTS: subset name

COMMENTS: This function lets you change names you entered with [M] (make new subsets) or with a spreadsheet program. To specify which subset names to change, type either a subset name, a name combined with a wild card, or just a wild card. Entry of a wild card gives a report screen telling you how many names match your specification. You can type [RET] to accept the matches or [ESC] to try again if no names were found. The function displays the name and number of each subset specified and lets you change it. Subset columns are numbered from left to right. The first subset after the PHRASE column is numbered 1.

All of the editing options can be used when you are editing a subset name. A subset name may be spelled with any printable ASCII characters and may have no embedded spaces. Do not begin a name with sharp (#) unless it's one of the KVS reserved sharp subset names (#edit or #voice_command). A subset name may contain up to 25 characters.

NAME: **[C] Copy a subset**

FUNCTION: Copies translations from a source subset into a destination subset.

PURPOSE: To augment a subset or merge translation subsets or create an all-inclusive subset containing every phrase in the language model.

ARGUMENTS: name of source subset, destination subset

COMMENTS: This function copies all translations from a subset (the source) into another subset (the destination). If the same phrase exists in both subsets and is translated differently, the source translation overwrites the destination. Both source and destination must exist before the copy command is given. If the destination does not exist, create it with [M] (Make new subset) before giving the [C] command.

To create an all-inclusive subset containing every phrase in the language model, give an answer of [^W] (wild card) to the question "Copy from which subset?, and name the all-inclusive subset when asked "Copy into which subset?" Each subset in turn will be copied over. If a phrase is translated in more than one subset, the translation in the highest numbered subset will end up in the destination.

To merge or consolidate subsets, use the subset you wish to discard as the source and, when the copy command is finished, delete the source with [R] (remove subset).

Phrase Book File Management

A choice of [F] (files) from the main <edit> menu gives the phrase book menu shown below. Individual choices are separately explained following the menu. To carry out a selection, <edit> must have the answers to a few preliminary questions. Because the window has only three lines, the display asks only one question at a time. After you supply an answer, the window changes and the next question is displayed. The function and purpose of each menu item are given below, together with the answers (arguments) you supply and a description of the operation.

PHRASE BOOK MENU		
[L] List phrase books	[D] Delete phrase book	[M] Merge phrase book
[S] Save phrase book	[P] Publish language model	[RET] for Main Menu

NAME: **[L] List phrase books**

FUNCTION: Lists the phrase book(s) stored on the defined <datapath> and <drive>. Lists all language models stored on the defined <drive> under \KVSLMOD.

PURPOSE: To find out which phrase books are available for editing and which language models are available for borrowing.

ARGUMENTS: phrase book name

COMMENTS: Specify which phrase book names to list by typing either the name of a phrase book, part of a name combined with a wild card, or just a wild card. An answer of a wild card alone will list all phrase books in your directory. Pressing [RET] after the display of phrase books will list all the published language models.

NAME: **[S] Save phrase book**

FUNCTION: Saves a phrase book to your user directory under the same name or a different name.

PURPOSE: To preserve editing changes. To copy a changed phrase book to disk under a different name.

ARGUMENTS: phrase book name

COMMENTS: You can change the name of the phrase book when you save, or you can use the same name. If you ask to save editing changes under the current name, <edit> warns that the version on disk will be overwritten. If you supply a different name <edit> gives two versions of the phrase book on disk: your original and an edited copy. Phrase book names may contain up to 8 printable ASCII characters except DOS reserved characters. The following characters are reserved by DOS and are illegal in phrase book names:

Invalid characters in phrase book names

. " \ / [ ] :

| < > + = ; ,

ASCII characters less than 20H

NOTE

The phrase book name "enroll" is reserved.
You cannot rename a phrase book to "enroll."

When a phrase book is written to disk, <edit> first condenses the language model component to save disk space if the condense LM switch is ON (see "Setup Menu, [C]"). You know a phrase book is being condensed when you see message such as

Condensing phrase book digits.
Saving phrase book digits.

Condensing slightly increases the time required to save. If a faster save operation is important, choose [S] (setup menu) from the main menu and turn off the condense LM switch. If you ask to save a phrase book you haven't changed, <edit> gives you the chance to abort or continue.

NAME: **[D] Delete phrase book**

FUNCTION: Removes a phrase book from the fixed disk.

PURPOSE: To recover space on the fixed disk by removing all files associated with the named phrase book from your directory (\KVSUSERS\{your name}). One phrase book uses between 150 and 250 Kbytes of disk space, depending on whether it has a unique voice profile and on the size of the language model component.

ARGUMENTS: phrase book name

COMMENTS: Specify which phrase book(s) to delete by typing either the name of a phrase book, part of a name combined with a wild card, or just a wild card. If the message "Found no match" is returned, the phrase book does not exist in your user directory.

As with any command that destroys data, [D] asks for confirmation before proceeding with the deletion. When asked to confirm, type [Y] to proceed with the deletion or [N] to keep the phrase book.

You cannot delete your ENROLL phrase book with this function.

NAME: **[P] Publish language model**

FUNCTION: Saves the language model from your phrase book to the systemwide language model directory, \KVSLMOD.

PURPOSE: To make a language model available to other users. Once a language model is published in \KVSLMOD, others can make a phrase book from it.

ARGUMENTS: language model name

COMMENTS: The name of the currently loaded phrase book is displayed for you to edit. Entry of a new name and [RET] copies only the language model files to the \KVSLMOD directory under the name. Entry of just [RET] copies the language model files to the \KVSLMOD directory under the phrase book name.

Before saving, <edit> checks \KVSLMOD to see if a language model by that name is already published. If it is, <edit> will not permit you to publish. You can publish the language under a new name if you like by changing the name specified.

If the condense LM switch is ON and changes have been made, the language model will be condensed before it is saved to disk. (See [S] (save phrase book).)

Valid language model names can have up to 8 characters and must follow the rules for file names permitted by MS-DOS. (See under [S] (save phrase book).)

NAME: **[M] Merge phrase book**

FUNCTION: Merges language models from two sources, making a new phrase book from an existing set of training tokens and a revised language model.

PURPOSE: To include changes from one phrase book in another; to augment a phrase book with selected subsets either from another phrase book or a public language model. To merge the spoken part of a phrase book (tokens) and a new version of the written part (language model).

ARGUMENTS: phrase book name, subset name

COMMENTS: In response to "Type phrase book name:" specify a source from which book to merge subsets by typing a phrase book name and [RET] or a language model name and [RET]. The source will be merged into the currently loaded phrase book (the destination). If the merge adds phrases, the new phrases must be trained when you exit from <edit>. Therefore, it's good practice to merge a small phrase book into a big one rather than the other way around. The merged phrase book may be saved to disk under the same name or a different name (see [S] (save phrase book) above). After you specify a source, [M] (merge) asks which subsets are to be merged ("which subset?"). An answer of just a wild card merges all subsets. An answer of a single subset name merges all unique phrase-translation pairs from the source phrase book into the destination phrase book. After merging phrase-translation pairs from one subset, you can continue merging pairs from other subsets or exit to the files menu.

If the same phrase-translation pairs are present in both source and destination, the source pairs replace the destination pairs. A pair that is present in the destination subset without a counterpart in the source phrase book is left unchanged.

To see the phrase books and public language models available for merging, type only a question mark at the "Type phrase book name:" prompt. To see the available subsets for merging, type "?" at the "which subset ?" prompt.

Tokens from the source phrase book are not merged. The setup menu item [T] (token number) for the destination phrase book determines how many tokens will be added for new phrases rather than the number of tokens that existed in the source phrase book.

[M] duplicates part of the function of KVMODPB. It adds and changes information but deletes nothing. (KVMODPB deletes phrases that don't exist in the source but do exist in the destination.)

Reporting Language Model Size and Contents

The functions described in this section display statistics about the size of the language model and list the contents of the language model.

A choice of [I] (information) from the main menu gives information on phrase book size. This function reports the number of phrases, the number of subsets, and how many tokens per phrase you have room to store in the currently loaded phrase book.

A choice of [V] (vocabulary list) from the main menu gives the "List What?" menu, shown below. This function lets you display the phrases you can say, with or without translations, in one or more subsets.

```
[I] INFORMATION ON PHRASE BOOK SIZE
```

```
VOCABULARY LIST: LIST WHAT?
[P] Phrases                          [S] Subsets
[T] Phrases, subsets and translations [RET] for Main Menu
```

NAME: **[I] Information**

FUNCTION: Reports information on phrase book size.

PURPOSE: To discover the maximum number of tokens you can record in training. To display the number of phrases and subsets in the phrase book. To display the token resolution.

ARGUMENTS: none

COMMENTS: The information window shows the number of subsets and phrases in the phrase book, the token resolution, and the number of tokens there is room to train.

NAME: **[P] List phrases**

FUNCTION: Lists phrases in the vocabulary.

PURPOSE: To remind an end-user of the available phrases to say. To remind a vocabulary designer of phrases in the application vocabulary.

ARGUMENTS: phrase

COMMENTS: Specify which phrase(s) to list by typing either a phrase, part of a phrase combined with a wild card, or just a wild card. Entry of a wild card alone gives all the phrases in the language model together with a count of how many phrases there are. Entry of a phrase combined with a wild card gives all phrases that match the specification together with a count of how many phrases in the language model match the specification.

Use wild cards to make requests such as "List all phrases beginning with b" or "List all phrases that have the sound ee." See Table 12.4 for use of wild cards to search a language model.

Phrases are listed in numerical order. Numbers are assigned as the phrases are typed into the language model.

NAME: **[S] List subsets**

FUNCTION: Shows names of all subsets in the language model.

PURPOSE: To remind an end-user or vocabulary designer of subset names. Subset names must be supplied with the command to change subsets and with various <edit> commands.

ARGUMENTS: subset name

COMMENTS: To specify which subset names to list, type the subset name, part of a subset name combined with a wild card, or just a wild card. Entry of a wild card alone displays names and numbers of all the subsets.

NAME: **[T] List phrases, translations, subsets**

FUNCTION: Lists one or more phrase-translation pairs in one or more subsets.

PURPOSE: To provide help messages for the end-user and the designer.

ARGUMENTS: phrase, translation string, subset name

COMMENTS: This powerful command lets you query the language model in many different ways. Sample queries are listed in Table 12.5.

TABLE 12.5 VOCABULARY LIST

To:	Answer the prompt	with
Display the language model	which phrase	^W
	which subset	^W
	which translation	^W
Find which translations are subset CAP	which phrase	^W
	which subset	CAP
	which translation	^W
Ask what phrase gives the translation "edit menu"	which phrase	^W
	which subset	^W
	which translation	edit menu
Find the translations for the star (*) phrases	which phrase	*^W
	which subset	^W
	which translation	^W
Find all subset switching commands	which phrase	^W
	which subset	^W
	which translation	^W<subset^W
Find all translations containing \r	which phrase	^W
	which subset	^W
	which translation	^W\r^W

Setup of Phrase Book for Data Collection

The functions described in this section are used to

- reset the number of tokens per phrase;
- reset token resolution;
- enable or disable condensing of the language model on phrase book saves;
- specify which voice profile to assemble with the phrase book.

PHRASE BOOK SETUP MENU		
[R] Resolution: 6	[C] Condense LM: ON	
[T] Tokens/Phrase: 3	[V] Voice Profile: digits	[RET] Main Menu

NAME: **[R] Resolution**

FUNCTION: Changes the amount of information stored in the KVS's internal image of the speech sound for each token made in training.

PURPOSE: To improve recognition accuracy or to free KVS memory.

ARGUMENTS: an integer from 1 to 7 (default is 6)

COMMENTS: Resolution designates the amount of data stored for each token made during enrollment and training. The higher the number, the more exact is the KVS's internal image of the speech sound; a more exact image can improve recognition accuracy, but it takes more memory. Increasing resolution by 1 increases memory use by 17%, and correspondingly decreases the maximum vocabulary size and the maximum length utterance the KVS can accept. For large vocabularies, it also reduces the number of tokens per phrase you can train.

If you change token resolution in an existing phrase book, you must train all the tokens again and may wish to compute a new phonetic model (see "Changing Token Resolution" and "Compute a Phonetic Model on the Application Vocabulary," Chapter 13).

CAUTION

At a resolution of 6, the maximum utterance duration (2.0 seconds) is reduced by 33%, as is the vocabulary capacity.

NAME: **[T] Tokens per phrase**

FUNCTION: Resets the number of tokens per phrase that will be prompted for during training.

PURPOSE: To increase recognition accuracy for the whole vocabulary by increasing tokens per phrase. To free KVS memory by decreasing tokens per phrase.

ARGUMENTS: a number between 1 and 100 (the default is 3)

COMMENTS:

The number you specify gives the number of tokens of each phrase to be trained in the future when you edit your phrase book or your tokens. If you change tokens per phrase in an already trained phrase book, the new number has no effect on past training. If you publish the language model after changing tokens per phrase, the current number of tokens for each phrase in the phrase book is carried over into the public language model. Use the [C] (count) function in <train> to find out how many tokens each phrase has.

Two factors determine how you set tokens per phrase: confusability of vocabulary phrases and vocabulary size.

(1) Make tokens per phrase a larger number if the vocabulary is confusable (and vocabulary size is not an issue) to get better recognition accuracy.

NOTE

Do not set tokens per phrase to 1 if you intend to make a voice profile on this vocabulary.

(2) For large vocabularies, the percent of token memory used becomes an issue. When token memory use is above 100%, and the average utterance is over .5 seconds, recognition response can slow. If the application vocabulary is large, set tokens per phrase to a small number and use the <train> command to add tokens for individual phrases that are confusable.

NAME: **[C] Condense LM switch**

FUNCTION: Enables or disables language model compression before writing a phrase book or a language model to disk.

PURPOSE: To save disk space and KVS memory.

ARGUMENTS: ON/OFF

COMMENTS: Setting the condense LM switch to ON causes the save phrase book function [S] to execute in two steps. The first step condenses the language model. The next step saves the phrase book to disk with the "Please stand by..." message. Turning off the switch speeds up the save phrase book function slightly.

NAME: **[V] Voice profile**

FUNCTION: Assembles a specified voice profile with the current phrase book or language model.

PURPOSE: To assemble a phrase book for training and recognition. Duplicates one function of KVMAKEPB.

ARGUMENTS: voice profile name

COMMENTS: The named voice profile files (.kam and .kpm) must already be present in your user's directory as a result of running KVENROLL on a target vocabulary.

It is not necessary to run this function on a newly created phrase book. When <edit> creates a new phrase book from scratch, it assembles the ENROLL voice profile files with the language model files by default.

WHAT YOU HAVE LEARNED

On completing this section, you've learned how to make, train, test and change a phrase book. The next part of this manual tells you how to improve KVS recognition accuracy, troubleshoot common performance problems, tailor start-up for end-users, and transfer language models between users.

CHAPTER 13
THROUGHPUT: TUNING THE KVS FOR BEST PERFORMANCE

CONTENTS:

- Error Prevention: Designing Errors Out
- Error Correction: Improving Tokens
- How to Minimize the Effect of Background Noise
- Talking Faster: Speed Up Digit Entry
- Voice Macros: Executing Several Commands with a Single Phrase
- Voice Correction
- Adjusting Parameters Controlling the Candidate List
- Output Modes: Customizing Output

This chapter discusses ways to increase the ease and efficiency of working by voice. It explains how to work quickly and keep working in case of errors. Several general areas of increased throughput are discussed:

- Error prevention: designing errors out
- Error correction: improving tokens (reference data)
- Talking faster: speeding up digit entry
- Voice macros: executing several commands with a single phrase
- Voice correction

Also, parameters controlling recognition and recognition output modes are discussed so that a language model designer can test the application with voice.

ERROR PREVENTION: DESIGNING ERRORS OUT

Accuracy is the measure of how well the KVS performs in an application. You can prevent some kinds of recognition errors by choosing phrases for your application carefully and by removing confusable phrases to different subsets.

Make Phrases More Distinctive

Errors caused by phrases that sound alike to the KVS can be prevented by choosing alternative phrases that are acoustically distinct and not easily confused. Chapter 11, "Find Problem Words," gives a procedure for finding confusable phrases in recognition candidate lists. When you find two phrases confused, change one of them. For instance, if the KVS confuses the phrases "edit" and "eight," performance will be improved by using the phrases "edit file" and "eight" instead. Two similar phrases, such as "move cursor left" and "move cursor right," can be made into three very distinct phrases:

"move cursor"
"left"
"right"

Replace very short phrases with longer phrases: "up," for example, can be replaced with "move up" or "cursor up." When you've changed a vocabulary, test recognition of new phrases to be sure they don't sound like existing ones.

Remove Confusion By Subsetting

You can also use subsets to separate potentially confusable phrases. If confusable phrases belong to different subsets, one member of the pair can be eliminated as a candidate for recognition, and correct recognition of the other phrase becomes much easier. If the KVS confuses "pin" and "ten," "line" and "nine," and "live" and "five," put the number phrases in a different subset, such as Digits. "Pin," "line," and "live" will not have translations in this subset and so will not be confused with the digits.

Another way to remove confusion by subsetting is to design a language model that limits the number of phrases that can be recognized at every point in the application and switches to a new subset after every recognition. Such a language model will first listen for subset 1, then automatically switch to subset 2, then to subset 3, etc. This type of language model lends itself to an application where items are entered in a fixed order, as in the following example.

Commodity traders buy and sell contracts for fixed amounts of commodities such as soybeans, platinum, and wheat. They must specify whether the transaction is buy or sell, the number of contracts, the commodity, and the exchange. The following language model example lets a trader say a sentence such as "Buy five wheat Kansas escape." After each phrase, the KVS switches to a new subset and listens only for the alternative phrases that can come next. The phrases "buy" and "five," normally confusable, can never be mistaken for each other because one is always eliminated as a candidate for recognition.

LANGUAGE MODEL EXAMPLE:

A	B	C	D	E
PHRASE	Action	Quantity	Commodity	Exchange
*Suffix	<subset quantity	<subset commodity	<subset exchange>	<subset action>
buy	buy			
sell	sell			
zero		0		
one		1		
two		2		
three		3		
four		4		
five		5		
wheat			wheat	
pork			pork	
kansas				kansas\r
chicago				chicago\r

Changing vocabulary phrases and subsets requires changes to the language model. You can make these changes with <edit> (see Chapter 12) or with a spreadsheet program. Remember that any change to the spreadsheet or .prn form of the language model requires you to recompile it with KVMAKELM, put it in the phrase book with KVMODPB, and then retrain the new phrases. Short changes are easier to make with <edit>.

ERROR CORRECTION: IMPROVING TOKENS

Performance can improve with the replacement or addition of extra reference tokens to match in recognition. A user getting poor recognition can add tokens of questionable phrases with the <train> command. This process is described in Chapter 10, "Add Tokens." Addition and replacement of tokens by voice command is shown in the language model examples in Chapter 8 (see "Add Tokens" and "Replace Tokens").

You can specify more tokens for each phrase or for all phrases when making a language model. This process is described in Chapter 12 ("Tokens_per_ phrase"), and in Chapter 9 ("KVMAKELM, [-t]").

For Highest Accuracy: Changing Token Resolution

Resolution specifies the amount of information stored with each token during enrollment and training. The higher the value of token resolution, the more exact is the KVS's internal image of the speech sound. Higher resolution is as if the KVS zoomed in on the image and produced a larger, clearer snapshot of each phrase.

A more exact image can improve recognition accuracy, but any increase in resolution will cause some increase in reaction time. Depending on the size of your vocabulary, therefore, it may be better to add tokens at the same resolution rather than increase resolution. Increasing token resolution from the default of 4 to 5 increases memory use by 17% and decreases both maximum vocabulary size and the longest utterance the KVS can recognize by 17%. Token resolution should be kept at 6 or below.

Resolution is specified in the language model and is fixed for a vocabulary (see Chapter 12, "Phrase Book Setup Menu, [R] Resolution," and Chapter 9, "KVMAKELM, [-r]").

IMPORTANT

You cannot change resolution for only some phrases. If you change resolution in a trained phrase book, you must train all tokens again and

compute a new phonetic model as explained below.

IMPROVING ENROLLMENT DATA

Errors also may be eliminated by improving the statistical models the KVS uses during recognition. The KVS builds three models during enrollment: one of background noise (the environment), an acoustic model of the user's voice, and a phonetic model of linguistically relevant distinctions. The acoustic and phonetic models together constitute the voice profile. Usually, they are made from the ENROLL vocabulary, a set of phrases designed to include all the sounds of English. One way to improve recognition is to add tokens to the ENROLL vocabulary and recompute the ENROLL phonetic model. Another way is to compute the phonetic model on the application vocabulary itself. The last way is to recompute both the acoustic and the phonetic models -- the whole voice profile -- on the application vocabulary. These methods are discussed below.

Improving the ENROLL Phonetic Model

Enrollment creates a voice profile containing mathematical models of your speech in two files: ENROLL.KAM (the acoustic model) and ENROLL.KPM (the phonetic model). The phonetic model captures the variability in how you say sounds within phrases. Since it is a statistical procedure, the more data you give it, the better model it can build. To augment the ENROLL phonetic model, go back and add up to 3 more tokens of each phrase. (A total number of tokens per phrase greater than 6 will probably not add helpful information.)

There are three steps to the procedure: (1) make a backup copy of your present ENROLL.KPM (optional), (2) add tokens, and (3) make a new phonetic model. The speaking and computation process takes about 2½ hours. Start it before lunch or at the end of the day.) Here's how to do it:

1. Back up ENROLL.KPM (optional): type, at C>

```
CD \KVSUSERS\{your name}
COPY ENROLL.KPM OLDENROL.KPM [RET]
```

2. Add tokens: type

```
<train enroll [RET]
[A]
*7 [RET]
```

Speak new tokens as prompted. Don't skip a phrase. When all tokens are collected, press [S] to save.

3. Make a new phonetic model: type

```
<phonetic enroll> [RET]
```

Now the KVS enters a computation phase. You can use MS-DOS during this time, but responses will be slow at times. The resulting file, ENROLL.KPM, will be written over your original in \KVSUSERS\{your name}.

The new phonetic model will improve recognition in existing, trained phrase books and in new phrase books you go on to create. The KVS automatically uses the new ENROLL.KPM in place of the old one in all phrase books that have an enrollment voice profile. You can always revert to the previous phonetic model if you prefer: Use the DOS Rename command to rename the new ENROLL.KPM to NEWENROL.KPM. Then rename the file you saved as OLDENROL.KPM to ENROLL.KPM.

Computing a Phonetic Model on the Application Vocabulary

If your application vocabulary has more than 200 phrases with 3 tokens of each, it may be better to compute a new phonetic model on the application vocabulary than on the ENROLL vocabulary.

If you've already trained your application, creating a new phonetic model doesn't involve speaking again. You use the <phonetic> command, and the KVS produces a better picture of how you speak phrases and how you say them differently from time to time.

IMPORTANT

> The number of phrases in the vocabulary is crucial: when an application has fewer than 200 phrases, you must use the ENROLL vocabulary instead. There must be at least 500 tokens in the application phrase book.

The <phonetic> command is useful if you have a large vocabulary already trained or when you add a lot of phrases to an existing trained vocabulary. In phonetic modeling, the object is to have as many different phrases and tokens of those phrases as you can. You must have at least 500 tokens in your phrase book, and these must contain at least two repetitions of more than 200 phrases. Before computing a new phonetic model, confusable phrases should be eliminated from the vocabulary with the procedure described in Chapter 11 ("Find Problem Phrases").

A new phonetic model should be computed from the best possible token data. Before recomputing the phonetic model you should replace noisy tokens. Delete tokens that the KVS incorrectly and frequently proposes as a match (intruders). Also, if you have trained a single phrase with very different pronunciations, make the two pronunciations into different phrases. For example, if "the" is a phrase and you have some tokens pronounced with a short "e" and

others pronounced "thee," edit the vocabulary to make two phrases "thee" and "the," and retrain the tokens for these phrases.

You can try recognition as soon as the computation phase completes. The KVS automatically swaps the new phonetic model for the old one in the application phrase book. You should notice an improvement in accuracy for the application right away. Recognition on other vocabularies, however, will not be affected.

To make a vocabulary-specific phonetic model, type

<phonetic {phrase book name}>

Then let the KVS compute for a while. Computation time increases as the number of tokens or phrases in your vocabulary increases. It can take from 20 minutes to several hours to complete. At the end of phonetic modeling, the status message is

"current command action is: phonetic complete"

You can also add tokens to a vocabulary to meet the 500-token minimum for a new phonetic model and then give the <phonetic> command. To do this, start with <train> and add at least two tokens per phrase, keeping the total number of tokens per phrase at 6 or below. See Chapter 10, "Add Tokens." After training, run <phonetic> on the phrase book.

Enrolling on an Application Vocabulary Instead of on the Enroll Vocabulary

This one-step procedure has some advantages over the usual two-step procedure of enrollment followed by training:

- The user says only the phrases useful for the application.
- Speaking time is reduced since only the application vocabulary is spoken.
- Errors are reduced because the acoustic and phonetic models are tuned to the application vocabulary.

If you're training new end-users and want to avoid having them go through two phases of speaking, they can enroll on the application vocabulary rather than on the ENROLL vocabulary. Enrolling will then double as training for the application. The application *must* have at least 200 phrases and you must train at least 3 tokens of each. This technique provides an application-specific voice profile.

There is one important drawback. If the application vocabulary used to enroll is missing certain sounds such as "b" and "d," and you later try to add the phrases with those sounds, say "bad" and "dad," the KVS may not be able to tell the difference between them. If you

introduce new sounds into the vocabulary after you make an application-specific voice profile, accuracy may be poor on phrases containing the new sounds and you will have to enroll again.

To run an application-specific enrollment, type

KVENROLL {user name} { phrase book name} [RET]

The enrollment procedure is the usual one (see Chapter 4) except that the phrases prompted for are those in the application vocabulary rather than the balanced ENROLL vocabulary.

After the computation following speaking is complete, you can try recognition, as described in Chapter 11. Training the vocabulary can be bypassed since it was done by the prompting prior to computation. If recognition testing reveals problem phrases, tokens of those phrases can be added, replaced, or deleted with <train> (see Chapter 10).

HOW TO MINIMIZE EFFECTS OF BACKGROUND NOISE

The KVS can recognize spoken phrases with high accuracy in noisy conditions. If you are speaking to the KVS in a place that has a high level of background noise, you should first understand three things:

1. The headset microphone is a close-talking, noise-canceling microphone. You should wear it close to the lips and in a consistent position. The closer the microphone is, the better recognition will be.

2. The microphone check models background noise, so users should repeat it often, especially when the level of constant background noise changes.

3. The KVS rejection <threshold> discards poor matches and withholds response to random sounds like phones ringing or doors slamming.

In Training

If you are going to use the KVS in a space that has a high level of background noise, take the following precautions when you record tokens for enrollment and training:

1. Place the microphone very close ($\frac{1}{4}$") to your lips, at the corner of your mouth.

2. Keep the microphone position consistent when you are recording tokens with the position it had in the microphone check.

3. Lower the gain to 3 or 2.

4. Speak loudly and distinctly in enrollment and training.

5. Be careful that noise isn't recognized as phrases during enrollment or training. You may have to use [B] (go back) frequently if noises are accepted as sample words during enrollment or training.

Watch the speaking-level lights. *Green* should be on, and *yellow* should light occasionally. *Red* should never be on. If the red light comes on, turn the gain down and redo all tokens recorded while the red light was lit. If you lower the gain setting from the default for training, you must remember to lower the gain each time you use the KVS for recognition and to speak loudly enough to keep the green light on when you speak.

In Recognition

If you recorded tokens in a quiet environment and now want to use the KVS in a noisy environment, take the following actions to compensate for the difference in background noise conditions:

1. Keep the gain at the default value (gain = 5).

2. Speak up.

3. Lower the rejection <threshold> to allow poorer scoring tokens onto the recognition candidate list.

A noisy environment gives poorer matches because the KVS hears your voice plus a lot of noise, but the stored patterns are of your voice with less noise or different noise. Also, you may speak differently in a noisy environment.

Don't raise the gain. If the red light comes on, the gain is already too high. At a high gain setting it is possible to make the speech signal so loud that it will produce a distorted pattern for the KVS and adversely affect recognition.

TALKING FASTER: SPEEDING UP DIGIT ENTRY

The digits 0-9 are short, primarily monosyllabic words that can be said faster, if desired, by changing <parameter wordgap>. This parameter determines the minimum pause time between phrases. Values range from 1 (the briefest pause) to 7 (the longest pause) with a default value of 5.

A wordgap of 5 establishes a pause requirement of 180 milliseconds (1/6 second) between phrases. Changing the value of wordgap by 1

changes the minimum gap duration by 30 milliseconds. A millisecond is a thousandth of a second.

Shorten wordgap if no phrase in the vocabulary has a long internal pause, as the phrase "hipboots" does. Wordgap values of 1 and 2 should be used with caution. The higher wordgap values for a word such as "hipboots" tells the KVS to treat it as one word even though there is a silence between "hip" and "boots." If a silence is longer than the wordgap duration (>180 millisecond), the KVS treats the utterance as two phrases.

To reset the value of WORDGAP, type

<param wordgap n>

where n is a value between 1 and 7. This parameter will retain the value you set even if you load a new phrase book. It reverts to 5, the default, only when you type <restart>, when a user with a different name signs on, or when you type <param wordgap 5>. See also the language model examples in Chapter 8, where *ENTER and *EXIT are used to set and reset <param wordgap> automatically for entry of numbers.

VOICE MACROS: EXECUTING SEVERAL COMMANDS WITH A SINGLE PHRASE

One of the great strengths of voice input is that one phrase can execute a whole sequence of commands or cause whole phrases or paragraphs to be typed on the screen. What does the KVS do when it hears speech? It not only types, it translates. Translation sends a sequence of up to 498 characters to the host computer and allows the user to say what he or she wants to get done without thinking of the keyboard commands necessary to execute the task. For each word or phrase, a character string translation is stored that represents what the application software would expect to see typed into the terminal keyboard. The translation can be a single keystroke or an entire sequence of commands. A phrase that executes a series of commands is a voice macro.

Saying "standard terms," for example, can translate to "Ship via UPS [RET] Net 30 [RET] FOB Company Name, Waltham MA 02154 [RET]." A single, short, easy-to-remember phrase can execute several long commands. (See Chapter 8, "Language Model Examples.") Each KVS user can invent personal macros, such as "read mail" or "send mail," and enter translation strings for them. Any keyboard macros can easily be entered as translation strings.

VOICE CORRECTION

To make voice input easier and more productive, you can build in error correction by voice to your language models. When an error occurs or you change your mind, you can quickly correct it by voice and keep working.

There are three KVS reserved phrases that let you correct errors in recognition and in speaking by voice command. They are the sharp phrases

- #DELETE LAST WORD
- #NEXT CHOICE
- #TRY AGAIN

When the KVS hears "delete last word," it backs up and erases the last phrase typed on the screen. The command can be repeated up to 10 times, backing up in text by up to 10 phrases.

When the KVS hears "next choice," it erases the last phrase typed and then displays the next best recognition candidate. The command can be repeated until the user has seen all the recognition candidates.

When the KVS hears "try again," it erases the last phrase typed and then listens for a new spoken phrase, excluding the phrase just erased as a recognition candidate. The KVS will never type the same thing twice when "try again" is spoken. The use of these phrases is illustrated in the language model examples in Chapter 8.

Corrections can also be made by your application program with voice commands that you provide. Many applications let you reenter data by pressing [ESC] or typing "exit." Some applications have an undo function that cancels the last action. You can add these functions to your language model so you can do them by voice.

ADJUST PARAMETERS CONTROLLING THE CANDIDATE LIST

The KVS <parameter> command takes several different kinds of arguments. Those that affect recognition candidate selection are described in this section. A complete list of parameters appears in Appendix B.

The recognition parameters affect how many alternative candidates to the recognized phrase will be sent to your application program and how they will be chosen. They should be adjusted in combination with the <output> command, which determines how recognition results are sent to the host PC (see below).

Table 13.1 gives the names and default values of the <parameter> arguments

TABLE 13.1 Parameters Controlling the Candidate List

Name	Default value
BDIST	150
BSCORE	500
MAXRATIO	10
MAXPHRASES	3

The settings of these arguments determine whether a phrase is qualified to enter the candidate list, the list of top candidates for recognition that you see in ordered_list output mode. If your application requests only the top candidate, that is, if you are in top_choice mode (the default) or in voice application mode, these commands affect only the operation of the #NEXT CHOICE command. But if your application uses ordered_list output mode, these commands control membership in the candidate set. In eliminating bad matches from the candidate set, the KVS applies the BDIST threshold test first, then BSCORE, then MAXRATIO, and, last, MAXPHRASES.

BSCORE and BDIST set minimum thresholds for distances and scores. BDIST removes alternatives from the candidate list whose distance is greater than {bdist}. Small distances are good matches and correspond to high scores. If the KVS appears deaf, raising BDIST may help, but if BDIST is raised, BSCORE should be lowered.

BSCORE removes tokens from the candidate list whose score is lower than {bscore}. <parameter bscore> is directly proportional to <threshold>. Whenever the recognition <threshold> is raised or lowered by 1 point, BSCORE is raised or lowered by 50 points and vice versa. The default values are <threshold 5> (<param bscore 500>). Changing BSCORE allows you to make finer adjustments to the recognition threshold if you wish.

MAXRATIO uses the ratio between the highest and lowest scores rather than the scores themselves to establish membership in the candidate list. For instance, with MAXRATIO set to 5, if the best candidate phrase has a score of 1000, only phrases that score above 200 enter the candidate list. The default MAXRATIO is 10, so phrases with scores as low as 1/10th of the best candidate are allowed into the candidate list.

MAXPHRASES sets the maximum size of the candidate list in ordered_list output mode. The list of recognition candidates will never be longer than MAXPHRASES. The KVS rejects candidates if there are more candidates than MAXPHRASES. (Increasing

MAXPHRASES may not increase the number of candidates you see because of the values of the other parameters.)

Top_choice mode uses MAXPHRASES = 1; only 1 candidate (the best scoring candidate) is returned.

You can set MAXPHRASES higher -- up to a value of 64. If the language model allows #NEXT CHOICE, you may want to set MAXPHRASES to a larger value to get more candidates.

To set a parameter, type the command, a space, the parameter name, a space, and a value. For example

```
<param bscore 500>
<param maxphrases 3>
<param maxratio 10>
```

A parameter setting remains in effect until a new user signs on or the <restart> command is given, in which case it reverts to its default value.

OUTPUT MODES: CUSTOMIZING OUTPUT

Chapter 11 discussed two of the KVS modes for returning recognition results: ordered_list output and top_choice mode. Two other output modes can be used for experimenting with recognition performance. They are voice application mode and confirm mode.

All output modes are set with the command that specifies the form and content of the KVS's response to recognized phrases:

```
<output {mode} {destination}>
```

where {mode} is top_choice (t), ordered list output (o), confirm (c), or voice application (v). The {destination} parameter directs KVS output to the typeahead buffer {t} or a voice application program {p}.

Voice application mode is set this way:

```
<output voice_application program>
```

It returns to a voice application program a list of all reference tokens scoring above the recognition threshold (BSCORE). Voice application programs are programs that you write that call on the KVS Run-Time Library for voice I/O functions (the Run-Time Library is an optional set of software routines). In voice application output mode, tokens are listed by token index number and by a distance measure. A small number for the distance measure indicates a close match. The list can be displayed on the screen or directed to a program by changing the destination argument. Arguments to the <output> command can be abbreviated to one letter as given above.

Ordered output mode is set this way:

<output ordered_list typeahead>

It types a list of translations for the top few candidates along with their scores. At the top of the list is the translation of the phrase that scored highest, followed by the second-place phrase, and so on. The list contains between 1 and n phrases, where n is the value of the parameter MAXPHRASES. The number of phrases that appear on the list is determined by the rejection threshold.

IMPORTANT

KVS <commands> embedded in the language model are executed *only* in top_choice mode or when sending the translation in confirm mode. The other modes do not send translations. They send only phrases, and no commands are executed.

Confirm mode can be thought of as an interrupted top_choice mode. It requires the user to intervene before a translation is sent to the host PC. This mode displays a list of between 3 and 5 candidate phrases in the KVS pop-up window. The best scoring phrase is ranked 1 and the poorest 5. The user presses [RET] or [1] to confirm selection of the top scoring phrase, or a key between 2 and 5 to confirm one of the other candidates. The key press tells the KVS which displayed phrase to translate for the host. The KVS then sends the confirmed translation to the host and sends any embedded KVS control commands to the KVS. Only the first 14 characters of each candidate phrase are displayed.

CHAPTER 14
TAILORING START-UP AND DISTRIBUTING LANGUAGE MODELS

CONTENTS:

- What Is an Autoexec.bat File?
- Automatic KVS Initialization
- Automatic KVS Start-up
- Automatic User Start-up
- How to Create a Batch File
- How to Store a Language Model and Transfer It Between Users

This chapter assumes familiarity with the workings of your PC's operating system, particularly with making and editing an AUTOEXEC.BAT file.

When setting up the KVS for an end-user, you may want to hide the details and streamline the operation. One way to do this is by customizing the end-user's environment. For example, you can set up an AUTOEXEC.BAT file so a user only has to switch on the PC to be signed on, with the KVS system software and a phrase book downloaded automatically. This chapter shows you how to automate KVS initialization, download, and start-up for each end-user. It also shows you how to give end-users private copies of public language models.

WHAT IS AN AUTOEXEC.BAT FILE?

In MS-DOS, a batch file is a file containing one or more DOS commands that executes when you type its name. The name of a batch file must include the extension .BAT.

AUTOEXEC.BAT is a special batch file that executes automatically when you start or restart your host PC. AUTOEXEC.BAT files must be in your root directory.

The file must contain at least the path command that tells the PC's operating system which directories to search to find the programs you want executed. By default, it should search the root and \KVPROG. If it can't find the program on the search path, it says "Bad command or file name."

The KVS INSTALL program adds the name C\KVPROG (the directory containing KVS executables) to the search path if the path command does not yet exist in AUTOEXEC.BAT. If the root directory has no AUTOEXEC, INSTALL creates one containing only the command PATH=C\KVPROG.

The path command must come before any other KVS command in the file.

AUTOMATIC KVS INITIALIZATION

The KVS is automatically initialized when you turn power on. When your PC is started, the C:\KVPROG line in AUTOEXEC.BAT leads the PC to search \KVPROG, and this in turn loads KV3DR.SYS, the KVS driver file that resides in your PC's RAM during operation.

AUTOMATIC KVS START-UP

From an AUTOEXEC.BAT file you can also assign a serial port to KVS communications, reset the baud rate for the communications

port, change the location of the KVS window on the screen, and download the KVS. A KVLOAD command in the AUTOEXEC file will download the KVS whenever you boot your PC. The KVS driver does window management without calls to the BIOS (BIOS off) by default. This can also be changed from AUTOEXEC.BAT.

To specify changes to the defaults, include the following commands in your AUTOEXEC.BAT file before KVLOAD (** are comments):

SET KVPORT=1	** Dedicates COM1 to KVS communications. COM2 is the default.
SET KVSPEED={BAUD}	** Sets the baud rate for the KVS communications port. Use this command to speed up phrase book transfers and system download time. A PC XT can have a baud rate of up to 28,800. A PC AT can have a baud rate of up to 38,400.
KVCNTRL window 22	** Relocates the start line of the KVS window to line n, where n is an integer from 0 to 22. The default is 22.
KVCNTRL bios ON	** Enables use of BIOS calls for window management.
SET KVSUSER=bill	** Assigns the default user subdirectory to \KVSUSERS\BILL. This name supplies the default argument to utilities such as KVMAKEPB and KVMODPB.
KVLOAD/N	** Downloads the KVS over the port specified by SET KVPORT except if the system is already loaded. The optional switch /N following the command word tells the KVS to avoid downloading if the system was already loaded. If the switch is set to /Y, the download will be performed regardless. If no switch is set and the system is already loaded, KVLOAD will ask the user what to do.

AUTOMATIC USER START-UP

When an AUTOEXEC.BAT file executes a KVLOAD and any time KVLOAD executes, any AUTOLOAD.BAT file in \KVSUSERS will also be executed. Remember that you can automate user start-up at boot-up time only if KVLOAD is in the AUTOEXEC.BAT file. AUTOEXEC.BAT runs first and calls KVLOAD, which then calls AUTOLOAD.BAT.

The use of AUTOLOAD.BAT is optional. The commands suggested here for an autoload file can also be sent from AUTOEXEC.BAT, typed from the keyboard, or written into a separate batch file as shown below (see "Automated User Start-up by Typing User Name").

From an AUTOLOAD.BAT file, you can send the KVS commands to sign on a user, download a phrase book for recognition, set a rejection threshold, and tell the KVS where to find the user's phrase books.

A second autoload file will also execute if the AUTOEXEC file includes the Set KVSUSERS command. For example, if AUTOEXEC sets KVSUSERS=BILL, then KVLOAD will execute a second autoload file if one is present in \KVSUSERS\BILL.

You can send any legal angle bracket commands from an AUTLOAD.BAT file. Some of them are the following (** are comments):

KVSEND {user bill}	** Signs Bill on as user.
KVSEND {threshold 4}	** Sets the recognition threshold to n, where n is an integer between 1 and 9. Include this command when the user gets better recognition with a different recognition threshold than 5, the default.
KVSEND {drive H:}	** Tells the KVS that the phrase books are on drive H rather than C, the default.
KVSEND {recognize newdates}	** Downloads the phrase book newdates and starts recognition.
KVSEND {param wordgap 4}	** Shortens (or lengthens) the required pause between words.
KVSEND {param listening ON}	** Enables recognition in the KVS window.

One KVSEND command can be used to send several KVS commands. To do this, enclose each KVS command in curly braces, for example:

KVSEND {user frank} {recog demo} {!choose \r\r\r}[RET]

This command signs Frank on, downloads his phrase book, starts recognition, and does a microphone check all by voice.

Automated User Start-up by Typing User Name

Another way to automate user start-up is to write a batch file named for the user, for example, BILL.BAT. Such a file is not invoked

automatically. Instead, it runs when you type the user's name at the C> prompt after a KVLOAD, for example:

```
KVLOAD [RET]
bill [RET]
```

This command causes BILL.BAT to execute. BILL.BAT can contain commands to sign on user Bill, start recognition of a customary phrase book, shorten the pause between words and turn on recognition in <edit>. Here is an example of BILL.BAT:

```
KVSEND {user bill} {recog pb} {param wordgap 4} {param listening on}
```

HOW TO CREATE A BATCH FILE

You can create or modify a batch file by using the line editor EDLIN which comes with MS-DOS or by using the COPY CON command. (See "DOS Version 3.10 Reference Manual," February 1985, p. 7-24, "Batch File Commands," and "The EDLIN Commands," Chapter 8. EDLIN commands are also summarized for your convenience in Appendix B of this manual.

AUTOEXEC.BAT files must be in the root directory of the system disk. The first AUTOLOAD.BAT file must be in C:\KVSUSERS. The second, user-specific autoload file must be in the default users subdirectory under \KVSUSERS, for example,

C:\KVSUSERS\BILL\AUTOLOAD.BAT.

HOW TO STORE A LANGUAGE MODEL AND TRANSFER IT BETWEEN USERS

As the developer of a new vocabulary, you have two ways to make the language model available to other users:

1. You can store (publish) a language model in \KVSLMOD on the fixed disk for other users of the same computer.

2. You can move the language model to floppy disk for transfer to another computer.

This section explains each procedure.

Storing a KVS Format Language Model on the Fixed Disk

If you developed a language model with <edit>, simply publish it and sign off. (See Chapter 12, "[P] Publish Language Model.") Tell the new user the public name of the language model, and let him or her sign on to train. Here's an example:

1. Bill wants to create a language model for the dash application and call it billdash. To start making the language model, he types:

   <edit billdash>

2. When Bill finishes entering and testing the vocabulary and translations, he returns to the main <edit> menu, presses [F] (files menu), and publishes the language model under the name dash:

```
in <edit>:      [F]          **choose files menu(in edit)
in <edit>:      [P]          **choose the publish function
in publish:     [CTRTL-C] **clear billdash as the default name
in publish:     dash [RET]   **answer "which phrase book"
in publish:     [CTRL-Z]     **return to main level <edit>
in <edit>:      [ret]        **exit <edit>
                <bye>        **exit kvs
```

   The language model is now saved in \KVSLMOD as dash and in \KVSUSERS\BILL as part of the phrase book billdash.

3. Mary wants to use the dash vocabulary. She signs on and types <train dash>. <train> gives Mary a copy of dash from \KVSLMOD, and assembles it with her enrollment voice profile to make a new phrase book ready for training. Here's what Mary types:

```
<user Mary>
<train dash>
[RET] [RET] [RET]      **microphone check
[A]                    **add tokens
[RET]                      **train whole vocabulary
```

   Mary will be prompted for tokens of the dash vocabulary phrases. When she finishes training, she'll save the tokens to disk and have a complete dash phrase book in \KVSUSERS\MARY. She can rename the phrase book if desired by using <edit> to save the phrase book to disk under a new name.

Transferring an ASCII Language Model via the Fixed Disk

If you developed an ASCII format language model with a spreadsheet program or a text editor, run KVMAKELM on the .prn file (see Chapter 9). Tell the new user the name of the public language model and let him or her sign on to train. Here's an example:

1. Bill creates a language model for the dash application with a spreadsheet program and calls it billdash.prn. He then converts the ASCII language model to KVS format by running KVMAKELM with the -l modifier ([-l] gives the output language

model the name dash instead of the name of the input file, billdash):

KVMAKELM -l dash billdash.prn

2. When KVMAKELM finishes, the language model is now saved in \KVSLMOD as dash. Now Bill can quit and Mary can sign on.

3. Mary wants to use the dash vocabulary. She signs on and types <train dash>. <train> gives Mary a copy of dash from \KVSLMOD and assembles it with her enrollment voice profile to make a new phrase book ready for training. Mary's procedure is the one described earlier. She signs on and types <train dash> (see "Transferring a KVS Format Language Model via the Fixed Disk," item 3, above).

   Notice that <train> does the work of KVMAKEPB, and lets the developer bypass running that utility.

Transferring a Language Model via Floppy Disk

After the language model files are stored in \KVSLMOD, a developer can transfer them to floppy disk by running the phrase book backup and restore utility, KVSMOVE. KVSMOVE followed by the argument /L, will back up and restore just the language model from \KVSLMOD. Follow these directions:

1. Format a blank floppy disk.

2. At the C> prompt, type

   KVSMOVE BACKUP dash \L [RET]

3. When prompted, insert a blank formatted floppy disk in drive A and press any key when you are ready. This command backs up the dash language model files (dash.klm and dash.kmp) from \KVSLMOD to floppy disk.

4. Label your backup disk with the name of the language model and date of the back p.

A new user transfers the language model to another computer by reading it from floppy to fixed disk. Again, the KVSMOVE command is used, for example:

KVSMOVE RESTORE {dash} \L [RET]

The Restore program writes the language model files to \KVSLMOD. Once the files are there, the user simply trains the vocabulary, as described above (see "Transferring a KVS Format Language Model via the Fixed Disk," item 3).

APPENDIX A PC-COMPATIBLE KEYBOARD CODES

KEY NAME	NORMAL	WITH SHIFT	WITH CTRL	WITH ALT
A	a	A	\001	\0\036
B	b	B	\002	\0\060
C	c	C	\003	\0\056
D	d	D	\004	\0\040
E	e	E	\005	\0\022
F	f	F	\006	\0\041
G	g	G	\007	\0\042
H	h	H	\010	\0\043
I	i	I	\011	\0\027
J	j	J	\012	\0\044
K	k	K	\013	\0\045
L	l	L	\014	\0\046
M	m	M	\015	\0\062
N	n	N	\016	\0\061
O	o	O	\017	\0\030
P	p	P	\020	\0\031
Q	q	Q	\021	\0\020
R	r	R	\022	\0\023
S	s	S	\023	\0\037
T	t	T	\024	\0\024
U	u	U	\025	\0\026
V	v	V	\026	\0\057
W	w	W	\027	\0\021
X	x	X	\030	\0\055
Y	y	Y	\031	\0\025
Z	z	Z	\032	\0\054
1	1	!		\0\170
2	2	@	\0\003	\0\171
3	3	#		\0\172
4	4	$		\0\173
5	5	%		\0\174
6	6	¢	\036	\0\175
7	7	&		\0\176
8	8	*		\0\177
9	9	(		\0\200
0	0	)		\0\201
-	-	_	037	\0\202
=	=	+		\0\203

APPENDIX A PC-COMPATIBLE KEYBOARD CODES

KEY NAME	NORMAL	WITH SHIFT	WITH CTRL	WITH ALT
F1	\0\073	\0\124	\0\136	\0\150
F2	\0\074	\0\125	\0\137	\0\151
F3	\0\075	\0\126	\0\140	\0\152
F4	\0\076	\0\127	\0\141	\0\153
F5	\0\077	\0\130	\0\142	\0\154
F6	\0\100	\0\131	\0\143	\0\155
F7	\0\101	\0\132	\0\144	\0\156
F8	\0\102	\0\133	\0\145	\0\157
F9	\0\103	\0\134	\0\146	\0\160
F10	\0\104	\0\135	\0\147	\0\161
HOME	\0\107	\0\3777G	\0\167	
↑	\0\110	\0\3778H	none	
PAGE UP	\0\111	\0\3779I	\0\204	
←	\0\113	\0\3774K	\0\163	
	none	\0\3775L	none	
→	\0\115	\0\3776M	\0\164	
END	\0\117	\0\3771O	\0\165	
↓	\0\120	\0\3772P	none	
PAGE DOWN	\0\121	\0\3773Q	\0\166	
INS	\0\122	\0\3770R	none	
DEL	\0\123	none	none	
PRT SC	none	\0\67		
- (minus)	\0\377-J	\0\377-J		
+ (plus)	\0\377 + N	\0\377 + N		
BACKSPACE	\b			
NEWLINE	\n			
TAB	\t	\0\017		
FORMFEED	\f			
RET	\r		12	
ESC	\e			
SPACE	\s			
backquote		176		
break			\0\106	
[		173	33	
]		175	35	
;		72		
apostrophe		42		
"beep"	<beep>			

APPENDIX B
COMMAND SUMMARIES

Section I of this Appendix lists the commands you can use at the MS-DOS C> prompt to perform generalized KVS tasks. These commands invoke utility programs that run on the IBM PC host. Commands and arguments can be typed in uppercase or lowercase.

Section II lists the name, arguments (if any), and purpose of every KVS <angle bracket> command.

Section III summarizes the <edit> commands.

Section IV summarizes EDLIN commands.

SECTION I. COMMANDS TO INVOKE HOST UTILITIES (at the C> prompt)

COMMAND	ARGUMENT	PURPOSE
INSTALL	none	Installs KVS system software from floppy disk to fixed disk. The command A:INSTALL reads the system software from a floppy in drive A onto the fixed disk. (See Chapter 2.)
KVCNTRL	OFF	Logically disconnects the KVS. (See KVRESYNC and Chapter 14.)
	STATUS	Reports KVS status information.
	WINDOW n	Relocates start line of KVS information window to n, where n is a number between 0 and 22 and 0 indicates the top of the screen. (See Chapter 14.)
KVENROLL	{your name}	Makes a directory for your phrase books, loads the KVS. Get ready to say the ENROLL vocabulary. (See Chapter 4.)

COMMAND	ARGUMENT	PURPOSE
KVLISTLM	{language model name}	Prints a language model by subsets or prints the vocabulary alone, alphabetically. Also converts an <edit> language model to spreadsheet format. (See Chapter 9.)
KVLOAD	/Y or /N	Downloads KVS speech recognition software over COM1 or COM2. (See KVPORT.) /Y means perform the download if the KVS is already loaded. /N means don't download if the system is already loaded. (See Chapters 2, 14.)
KVMAKELM	{language model name} (ASCII format)	Produces a language model in KVS format from a language model in ASCII format (.prn file). (See Chapter 9.)
KVMODPB	{your name} {language model name}	Merges the revised language model with an already trained phrase book after you have changed words in a vocabulary and run KVMAKELM. (See Chapter 9.)
KVSMOVE	{backup/restore} {your name} {pbook name}	Copies an entire phrase book to or retrieves one from floppy disk. (See Chapter 4.)
	/L	Copies just the language model to floppy disk. (See Chapters 2, 14.)
KVRESYNC	none	Logically reconnects the KVS and host (see KVCNTROL OFF). (See Chapter 2.)
KVSEND	{angle bracket command}	Sends any text to the KVS from a batch file.
KVTRAIN	{your name} {pbook name}	Loads phrase book, gets it ready for you to train. (See Chapter 14.)
SET	KVPORT={n}	Dedicates a host COM port to KVS communications, where n is 1 for COM1 and 2 for COM2. (See Chapter 2, 14.)

COMMAND	ARGUMENT	PURPOSE
SET	KVSPEED={n}	Sets the baud rate controlling phrase book transfers, where n may be 28,800 for a PC XT or 38,400 for a PC AT. (See Chapter 14.)
	KVSUSER={name}	Makes \KVSUSERS\name the default user subdirectory, where name is your user name. (See Chapters 9, 14.)
UNINSTALL	none	Removes all KVS files except phrase books from your INSTALL disk.

SECTION II. ANGLE BRACKET COMMANDS FOR KVS CONTROL

The commands in this section, delimited by the brackets < >, are commands that control the operation of the KVS. They may be typed at the keyboard, spoken, or issued from an application program through the use of KVS Run-Time Library routines.

As soon as the KVS receives a <, a small window appears on the host's screen. The command being typed appears in the window, and when the KVS receives the closing delimiter (> or [RET]), execution begins. Spoken KVS commands are recognized because they have command strings as their translations. For example, a user can log off by saying "log off" if the language model translates the "log off" as <bye>.

All commands you type to control the KVS start with an open angle bracket character < and end with a close angle bracket > or a return [RET]; > and [RET] are the same to the KVS.

You can abbreviate commands by typing at least the first three letters of the command word in uppercase or lowercase. Case changes are not significant. Each argument following a command word is preceded by a space. Extra spaces are ignored. When an argument takes a value, the value is also preceded by a space; for example, <parameter bscore 500>.

If you omit the argument from a < command that takes an argument, the current value of the argument is displayed. For instance, if you type <user>, the KVS window will show the name of the user who is currently signed on.

To see the available commands, type <?>. To see all the available arguments to a command or all the available values for an argument type ? following the command word. For example, <param ?> names all the parameters you can set. <recog ?> names all the phrase books you can use for recognition.

A command that takes a phrase book name as an argument will accept a period in its place. A period signifies the name of the currently loaded phrase book.

COMMAND	ARGUMENT	PURPOSE
<?>	none	Shows available KVS commands at this point.
<ACOUSTIC>	{voice profile name}	Makes the acoustic model for a phrase book. Example: <Acoustic digits>
<BANNER>	none	Sends the KVS logo screen to the pop-up window.
<BEEP>	none	Makes the host PC beep once.
<BYE>	none	Signs off the present user.
<!CHOOSE>	{string}	Provides answers to KVS prompts or menu selections in <edit> or <recognize> or <train>. Used in a language model or a batch file rather than from the keyboard. If {string} is omitted, command clears the buffer. Example: <!choose a><edit .>
<DATAPATH>	{path name}	Changes the default datapath to the directory containing your phrase books. The datapath consists of a drive specification followed by a directory path. The default datapath is C:\KVSUSERS\{your name}. A <user> or a <restart> command resets the datapath to the default. Example: <drive C>
<DRIVE>	{device}	Changes the drive specification in the data path. The device is specified by a letter followed by a colon. The default drive is C. Example: <drive C>

COMMAND	ARGUMENT	PURPOSE
<HELP>	none	Lists the available commands at this point.
<EDIT>	{pbook name} or {language model name}	Adds/replaces/deletes phrases, translations, and subsets. Creates phrase books, merges two phrase books, saves phrase books to disk. Merges editing changes into training. Sets data-collection attributes of a phrase book. Publishes language models for other users. Examples: <edit dates>, <edit .>
	?	Show phrase books and language models available for editing. Example: <edit?>
<MICROPHONE>OFF		Virtual microphone OFF switch: disables sending of translations to the host PC after recognition. (#listen to me enables sending of translations after <microphone off>.)
<OUTPUT>	{format}, {destination}	Specifies a format and a destination for KVS output. Enter arguments separated by a space; for instance, <output o t>. Format may be any of: t=top_choice c=confirm_recognition o=ordered_list of translations v=voice_application list of phrases recognized m=measure_mode Destination may be any of: t=typeahead buffer (the usual user's destination) p=voice application program Examples: <output t t> <output o t>

COMMAND	ARGUMENT	PURPOSE
<PARAMETER>		Enter name of argument, a space, and integer or text value; for instance, <param brackets [ ]> or <param bscore 400>. To show the current value of a parameter, type the argument without a value; for example, <param brackets>.
	?	Shows list of settable parameters
	bdist {n}	Removes tokens from answer set whose distance score is > n, where n must be greater than zero.
	bscore {n}	Removes tokens from answer set whose score is < n, where n is between 300 and 700. See <thresh>.
	listen {ON/OFF}	Turns ON (or OFF) recognition in <edit> and KVS angle bracket commands. Enables (or disables) use of sharp subsets for translation.
	maxratio {r}	Rejection criterion used to reduce the list of recognition candidates (perplex set). Used in the expression that gives the bottom score for candidates in the perplex set: b/r=w where best score (b) is divided by maxratio (r) to determine worst score (w). If b=1000 and r=10, no words scoring below 100 enter the candidate list.
	wordgap {n}	Changes the minimum time to pause between phrases, where n is an integer between 1 and 7. A wordgap of 5, the default, requires a 1/6-second (180 msec) pause between words. Increasing the setting of wordgap by 1 increases the time to pause by 30 msec. Train with the same value of wordgap that you plan to use for recognition.

COMMAND	ARGUMENT	PURPOSE
<PARAMETER>	brackets {a e}	Redefines the < and > characters that delimit KVS control commands. "a" is the start character; "e" is the end character.
	quote char {q}	Defines a character to precede the bracket delimiters and any other character that normally terminates an angle bracket command ([ESC], [RET]). Tells the KVS command interpreter to use the next character literally instead of in its special, reserved sense. Backquote [`] is the default quote character.
	maxphrases {n}	Sets the maximum size of the list of recognition candidates (answer set). n is an integer between 1 and 64. For confirm output mode, maxphrases is usually 3, 4, or 5.
<PAUSE>	none	Turns off the microphone; allows time out from speaking.
<PHONETIC>	{voice profile name}	Computes the phonetic model for a phrase book. Examples: <phonetic dates>, <phonetic .>
<PROFILE>	{pbook name}	Starts prompting for a vocabulary-specific voice profile. <PROFILE> <ACOUSTIC> and <PHONETIC> make up KVENROLL. <acoustic> and <phonetic> must follow this command to complete a voice profile. Examples: <profile dates>, <profile .> **Warning**: Do not give the command <profile enroll>. Existing phrase books based on your enrollment will be invalidated.
<RECOGNIZE>	{pbook name}	Recognize all phrases included in the named phrase book. Examples: <recog dates>, <recog .>

COMMAND	ARGUMENT	PURPOSE
<RECOGNIZE>	?	Shows phrase books available for recognition.
<RESTART>	none	Restart the KVS operating system from its post-KVLOAD state. No user is signed on, no command is active, no phrase book is read in. Parameters are reset to default values.
<STATUS>		In RECOGNIZE shows whether the microphone is on. In ENROLL shows elapsed time since computation phase began. Status reports are issued after completion of current command unless you type <!STATUS>, which reports immediately.
<SUBSET>	{name}	Uses the named subset for recognition where name must be spelled exactly as it appears in the language model.
	?	Shows names of all subsets in language model. Examples: <subset command> <subset ?>
<THRESHOLD>	{n}	Sets bottom score (bscore) for recognition according to n. n is a number from 1 to 9. See also <param bscore>. Example: <threshold 4>
<TIME>	none	Displays time of day. Time is reported after completion of current command. Type <!TIME> to report time immediately.
<TRAIN>	{pbook name}	Records tokens for all vocabulary items in the named phrase book. These form the reference set used in recognition matching. Examples: <train newdates>, <train .>
	?	Shows phrase books available for training.

COMMAND	ARGUMENT	PURPOSE
<USER>	{your name}	Signs you on. Sets up a data directory for your phrase books if you are a new user; otherwise sets up the datapath to your directory.
	?	Shows all users with subdirectories under \KVSUSERS. Examples: <user bill>, <user ?>
<VERSION>	none	Displays KVS operating system software revision number (for troubleshooting).
<!WAIT>	{n}	Waits n msec before sending recognition result to host, when n is the number of milliseconds to delay, between 1 and 200. Example: <window this> <!wait 1000> <window will appear in slow motion>
<WINDOW>	message	Pops up the KVS information window with the message text displayed. Usually embedded in language models rather than issued from the keyboard. <window> without text erases the window. Example: <window the microphone is on>

SECTION III. <EDIT> COMMAND SUMMARY

MAIN MENU ITEMS

[A]	Add Phrase
[E]	Edit Phrase
[D]	Delete Phrase
[V]	Vocabulary List Menu
[F]	Files Menu
[RET]	Training/Recognition /Exit
[N]	Next Menu
[M]	Make New Subset
[R]	Remove Subset
[S]	Setup Menu
[I]	Information

[C]	Copy Subset
[X]	Change Subset Spelling

VOCABULARY LIST MENU

[P]	List Phrases
[S]	List Subsets
[T]	List Phrases, Subsets and Translations
[RET]	Exit to Main Menu

FILES MENU

[S]	Save Phrase Book
[L]	List Phrase Books
[D]	Delete Phrase Book
[M]	Merge Phrase Book
[P]	Publish Phrase Book
[RET]	Exit to Main Menu

SETUP MENU

[R]	Resolution
[T]	Tokens Per Phrase
[V]	Voice Profile
[C]	Condense Language Model
[RET]	Exit to Main Menu

SPECIAL CHARACTERS

[^O]	Show special options and cursor control characters
[^W]	Wildcard
[ESC]	Pop up one level toward main menu
[^Z]	Zap to main menu
[TAB]	Echo default string for the situation
[^S]	Choose subset for Add/Edit Phrase
[?]	Show available choices
[‘]	Enter next character as an octal code
[^V]	Toggle Mic ON/ OFF
[BACKSPACE]	Delete last character
[^C]	Clear string
[^R]	Restore original string
[^F]	Move cursor forward one character
[^B]	Move cursor back one character
[^A]	Move cursor to start of string
[^E]	Move cursor to end of string
[^K]	Kill from cursor to end of string
[^U]	Kill string up to cursor
[^D]	Delete character after cursor
[^P]	Put string in storage
[^G]	Get string from storage
[^N]	Next page (move cursor)
[^L]	Last page (move cursor)

SECTION IV. EDLIN COMMANDS

Command	Purpose
{line number}	Edits the line specified by line number. Enter a decimal integer from 1 to 65529. Period (.) means current line. Sharp sign (#) means the highest line number + 1.
A	Appends lines from a disk file to the file in memory.
C	Copies lines from one location to another in the file. [first line],[last line],[destination line],[count]C.
D	Deletes lines. [first line],[last line]D; or [line]D
E	Ends an editing session and writes the file to disk.
I	Inserts lines into the file before [line]. [line]I - CTRL + C to EXIT insert mode
L	Lists all lines in file being edited. [first line],[last line]L
M	Moves lines from one location to another. [first line],[last line],[destination line]M
P	Lists the file one page at a time. [first line],[last line]P
Q	Ends an editing session without saving the file.
R	Replaces string 1 in the search range with string 2. [first line][,last line]R<string1><ctrl-Z string2>
S	Searches for specified strings. [first line][,last line]S<string>
T	Transfers text from a file on disk to a specified location in the file being edited. [line]T[pathname]
W	Writes a specified number of lines (starting with line 1) to disk. [n]Write

APPENDIX C

HARDWARE AND SOFTWARE COMPATABILITY

COMMUNICATION BETWEEN THE KVS AND A MAINFRAME

IBM PCs or compatibles, when used with a KVS and communications accessories, can be voice-activated terminals. There are two methods for configuring a voice terminal: (a) using a software terminal emulator (such as CROSSTALK) and one serial line or (b) using 3270 terminal emulation for IBM mainframes (such as the Irma board).

CHECKLIST OF CRITICAL ITEMS FOR COMPATIBILITY

To make sure that your IBM PC XT, AT, or compatible will work correctly with the KVS, check the following list:

1. Does your system contain an unused communications port on an IBM Asynchronous Communications Adapter or equivalent? The adapter **must use** the INS8250 communications chip. (On Olivetti, use COM1 if in doubt.)

2. Is the unused serial port COM1 or COM2? If both are present, KVS addresses COM2. However, if there is only one communications port, the KVS addresses either COM1 or COM2. To address COM1 when COM2 is present, SET KVPORT=1 before giving the KVLOAD command.

3. Does your system contain an IBM or equivalent monochrome or color graphics display adapter? A non standard display adapter must use the Motorola MC6845 CRT controller chip.

   The KVS software writes directly to the screen, bypassing the BIOS, at 0xB000 or 0xB800 (hexadecimal). If your system has both kinds of displays, the KVS will address the monochrome screen. If your system uses the color graphics display adapter, it must be set to 80-column text mode. 40-column will work, but the rightmost part of the KVS window will not be visible on the screen.

4. Does your system run under PC-DOS or MS-DOS version 2.0 or later? Earlier versions will not support the capabilities of the KVS.

5. Does your system use any programs that preempt the following IBM interrupts?

   - Communications interrupt: 11 (COM2) or 12 (COM1)

APPENDIX C

HARDWARE AND SOFTWARE COMPATABILITY

COMMUNICATION BETWEEN THE KVS AND A MAINFRAME

IBM PCs or compatibles, when used with a KVS and communications accessories, can be voice-activated terminals. There are two methods for configuring a voice terminal: (a) using a software terminal emulator (such as CROSSTALK) and one serial line or (b) using 3270 terminal emulation for IBM mainframes (such as the Irma board).

CHECKLIST OF CRITICAL ITEMS FOR COMPATIBILITY

To make sure that your IBM PC XT, AT, or compatible will work correctly with the KVS, check the following list:

1. Does your system contain an unused communications port on an IBM Asynchronous Communications Adapter or equivalent? The adapter **must use** the INS8250 communications chip. (On Olivetti, use COM1 if in doubt.)

2. Is the unused serial port COM1 or COM2? If both are present, KVS addresses COM2. However, if there is only one communications port, the KVS addresses either COM1 or COM2. To address COM1 when COM2 is present, SET KVPORT=1 before giving the KVLOAD command.

3. Does your system contain an IBM or equivalent monochrome or color graphics display adapter? A non standard display adapter must use the Motorola MC6845 CRT controller chip.

   The KVS software writes directly to the screen, bypassing the BIOS, at 0xB000 or 0xB800 (hexadecimal). If your system has both kinds of displays, the KVS will address the monochrome screen. If your system uses the color graphics display adapter, it must be set to 80-column text mode. 40-column will work, but the rightmost part of the KVS window will not be visible on the screen.

4. Does your system run under PC-DOS or MS-DOS version 2.0 or later? Earlier versions will not support the capabilities of the KVS.

5. Does your system use any programs that preempt the following IBM interrupts?

   - Communications interrupt: 11 (COM2) or 12 (COM1)

- Keyboard interrupts: 9 and 16
- DOS function call interrupt: 33
- Timer interrupt: 28
- Equipment check: 17

The following is a partial list of incompatible standard programs for the IBM PC:

- XyWrite
- Visidex
- Visifile
- Visitrend
- Time manager
- Word vision
- Hayes Smartcom

(See note on dBASE III under "Bugs in DOS 2.0-2.1 For The IBM XT," below.) And some hardware that has been found to be incompatible:

1. METAFILE is a hardware box that plugs into the host's KVS serial port uses hardware copy protection but allows pass-through of data. The KVS requires a dedicated serial port and cannot share with devices such as this.

2. ATT 6300 PLUS requires a separate serial card such as an IBM half-card serial interface. COM1 does not appear to work.

POTENTIAL KVS CONFLICTS

Host application software and KVS driver software can make conflicting demands on the host operating system in two areas: communications and keyboard. This section describes how conflicts manifest themselves and what you can do to resolve them.

Communications Conflicts

The KVS requires use of a dedicated serial port. If it's connected to COM 1 and host application software initializes COM1, a conflict can result. The conflict manifests as no response to typing, followed by a

beep, followed by resumption of the ability to use the keyboard, but then there is no response from the KVS.

For example, you type on the keyboard and get no response, and 3 or 4 seconds later the PC beeps, and, after a brief (3-second) interval, the keyboard comes back. Then the KVS does not respond to talking or typing.

The easiest way to fix this is to put the KVS on COM2. If COM2 is absent or in use, try to reestablish KVS communications and keep on running. This approach will succeed if

1. the host application communicates over COM2;

2. the host application initializes COM1 only when it starts up and then doesn't address COM1 again;

3. you can suspend operation of the host application with a "hot key" while you restart the KVS.

This approach will not succeed if you have to restart the host application program because whenever it restarts, it initializes COM1 again, and the problem will repeat.

Read your program documentation to see if there is a "hot key" that suspends program operation to let a user talk to DOS. XTALK, for example, uses [ESC] followed by typing RUN to suspend operation. When you type EXIT to DOS, XTALK resumes.

Other programs such as SMART and IRMA also use "hot keys" to allow communication with the operating system. See your program documentation.

After using the "hot key" to suspend operation of the host application, reestablish KVS communications by typing at C>

```
KVLOAD/N
KVRESYNC
```

Then restore program operation by using the "hot key" or EXIT again, and return to your program. As long as you don't restart the program, the conflict is resolved.

Here is what you type, for example, to resolve the conflict with XTALK:

```
[ESC]
RUN
KVLOAD
KVRESYNC
EXIT
```

Keyboard Conflicts

Some programs initialize the DOS keyboard interrupts INT 9h and 16h when they start up. If this happens, the KVS driver software may be unable to insert recognition results into the typeahead buffer or to respond to <angle bracket> commands. The conflict shows up as a deaf KVS. When you speak, no translations appear on the screen. If you exit the host application and return to DOS, the translations suddenly pop to the screen. For each conflicting program a workaround must be designed individually.

IRMA

A patch available from Kurzweil AI allows DCA's IRMA 3270 terminal emulator to run with the KVS. Without the patch, you can't use IRMA and the KVS at the same time.

FORTE

To run with FORTE, type KVCNTRL FORTE ON. This allows FORTE PJ789 compatibility. This workaround requires you to run FDSCONF and set option 10 on the FDSCONF menu (sidekick or INT10). To disable the workaround, type KVCNTRL FORTE OFF and reset Option 10.

Host application programs that address the DOS keyboard buffer directly will not work with the KVS because the KVS driver allocates a new keyboard buffer. Such programs will not accept speech.

BUG IN DOS 2.0 - 2.1 FOR THE IBM XT

A program that fixes a bug in DOS 2.0 and 2.1 is supplied in \KVPROG on the INSTALL disk. The INSTALL program will install DMA64 only if you're using version 2.0 or 2.1 of DOS.

The bug, which can cause the DOS FORMAT command to fail, involves the inability of the XT's DMA system to deal with transfers across 64Kbyte boundaries, along with the FORMAT program's compliance with erroneous BIOS documentation. Depending on what programs are resident, what extensions to DOS are loaded in by CONFIG.SYS, and what version of DOS you're running, FORMAT may fail. Also, dBASE III may not work correctly without DMA64 installed.

If FORMAT fails, you'll see the message

Track 0 Bad, Format Failed

If you don't see this message, there is no problem.

The Kurzweil AI program is called DMA64. It checks for sufficient head room below the nearest 64K boundary, and if there isn't enough

it soaks up the space up to that boundary, effectively creating a 64K DMA region above the current limit.

The program checks whether loading another program with a buffer for acccessing your diskette would cause a problem. If it would, it tells you how many bytes it will use up to correct the problem and then goes ahead and corrects it. The maximum number of bytes DMA 64 will use to correct the problem is 9 x 512 bytes. Put a call to DMA64 at the end of your AUTOEXEC.BAT file after other programs have executed.

APPENDIX D
The KURZWEIL VOICETERMINAL

INTRODUCTION

The Kurzweil Voiceterminal (KVT) is a speech recognition system that sends recognized spoken words or commands to application programs running on a mainframe computer or a PC. Use of the KVT requires minimal knowledge of computers on the part of a user. Once the system is trained to a user's voice, the KVT recognizes spoken words or phrases and translates them into strings of characters which are interpreted by the host system as keyboard input.

When the host system is a mainframe computer, the KVT is configured either as an IBM/3270-compatible terminal or a VT-100 ASCII-compatible terminal. The ability to speak to a remote computer via the KVT has applications in data entry, data retrieval and command and control. The KVT can be connected to large host systems without any modification to the host hardware or software.

THE THREE KVT CONFIGURATIONS

1. The stand-alone KVT comprises three units cabled together: (1) an IBM PC-XT™-compatible computer, (2) a voice recognition unit, and (3) a monochrome display. The stand-alone system lets users speak to application programs running on the PC.

2. As an IBM/3270-compatible terminal, the KVT also comes with an IRMA™ board for communications to an IBM mainframe computer, and with IRMA software pre-installed.

3. As a VT-100 (ASCII)-compatible terminal, the KVT comes with CROSSTALK™ software for communications to an ASCII-based mainframe computer.

All three configurations have the 1,000-word recognition capability of the Kurzweil Voicesystem built in.

The computer comes with a 4.7 mHz. 8088 processor, 640K of RAM, a 360KB floppy disk drive (double sided/double density), and a 10MB hard disk. Two 25-pin serial ports and one parallel port are standard. It has one expansion slot (long board), occupied by the IRMA board in the 3270-compatible KVT. In the stand-alone PC version and in the ASCII version, this slot is empty and can be used for your own card -- a color graphics adaptor, for example. One serial port is dedicated to communications between the computer and the voice unit.

The KVT comes with MSDOS™, BASIC, and the Kurzweil Voicesystem software already installed on the hard disk. In addition, the 3270-compatible KVT has IRMA software and KVS IRMA compatibility software installed on drive C: in directory \IRMA. The ASCII-compatible version has CROSSTALK™ software installed on drive C: in directory \XTALK. An AUTOEXEC.BAT file is pre-

installed that sets the search path for data files, the format of the DOS prompt and downloads KVS software.

To modify the AUTOEXEC.BAT file or any of the pre-installed software, you will need to supply a text editor, or use the MS-DOS COPY CON procedure. This procedure, described on page 5-11 of the *WYSEpc User's Guide*, copies text from the keyboard into a named file.

The *WYSEpc User's Guide* covers basic operation of the KVT as a PC.

TRAINING AND USING THE KVT FOR SPEECH RECOGNITION.

The Kurzweil Voicesystem User's Manual covers every step of enrolling and training on the KVT, and using a vocabulary for speech recognition. It also describes how to make a vocabulary for your own applications. This KVT supplement replaces Chapter 2 in the Manual, *Installation and Startup*. It describes unpacking, and setting up the KVT.

To use the KVT for speech recognition, complete the hardware installation described below. Then go to Chapter 3 of the User's Manual, or follow the Quick-Start steps on the *white* Ready Reference Card.

UNPACKING INSTRUCTIONS

The KVT is packed for shipment in three cartons: one contains the computer, another the monochrome display, and the last the voice unit (KVS) and accessory parts. Unpack the cartons, and inspect all parts for shipping damage. Inventory the accessory parts and find the following:

- A plastic bag containing 3 cables:

  A PC power cord

  A voice unit power cord

  A KVS/Host communications cable

- A headset microphone
- A binder containing the KVS User's Manual
- A backup copy of the KVS System software contained on 4 floppy disks
- A backup copy of the WYSE PC Operating System software (MS-DOS) contained on 2 floppy disks
- A backup copy of the IRMA software contained on 4 floppy disks (KVT-BA)
- A backup copy of CROSSTALK software contained on 1 floppy disk (KVT-AA)
- Documentation: The Wyse PC User's Manual, a BASIC Manual, and, if ordered, an IRMA MANUAL or a CROSSTALK Manual.

SETTING UP THE KVT

Set up your KVT with the following physical requirements in mind :

Access

Clear a desktop area of at least 20" X 25" for the KVT. Inside the voice unit is a cooling fan vented through the left side of the cabinet. To maintain proper air flow, place the KVT on a hard surface and don't cover the vents.

Power Requirements

The KVT requires a standard 110VAC grounded outlet and 50 or 60 Hz current.

Assembly Instructions

1. Stack the 3 units on top of each other:
   - Place the voice unit on the desktop, rubber feet down, with the shiny metal alignment guides on top.
   - Place the computer on top of the voice unit: fit the rubber feet on the bottom of the computer into the alignment guides (see Figure KVT-1).
   - Place the display on top of the computer.

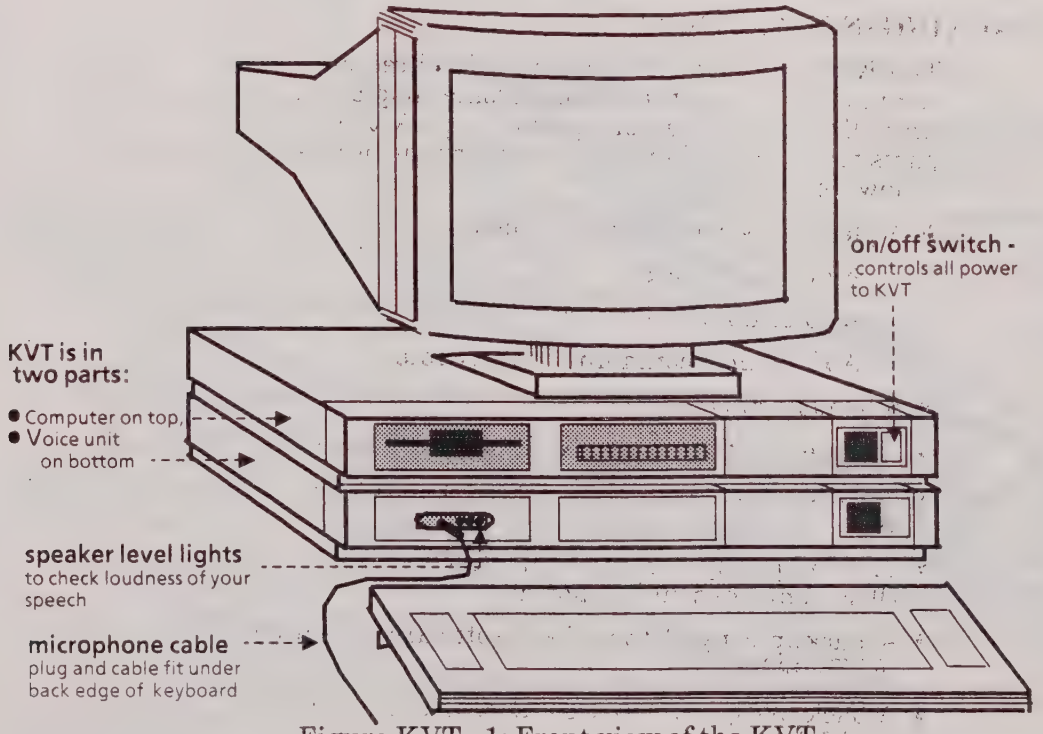

Figure KVT - 1: Front view of the KVT

2. Place the keyboard in front of the assembled KVT.
3. Next, before turning power on, hook up the three power cables, referring to Figure KVT-2:
   - Plug the power cord from the display into the (*female*) top connector on the voice unit (*lower* module).
   - Connect the voice power cord between the (*male*) bottom jack on the voice unit and the (*female*) top connector on the computer (*upper* module).
   - Connect the computer power cable between the (*male*) bottom jack on the computer and the wall.

Check the cable connections:

- Is the display power cord connected between the display and the the voice unit (top connector)?
- Is the voice power cord connected between the computer (top connector) and the the voice unit (bottom jack)?
- Is the computer power cord connected between the computer (bottom jack) and the wall?

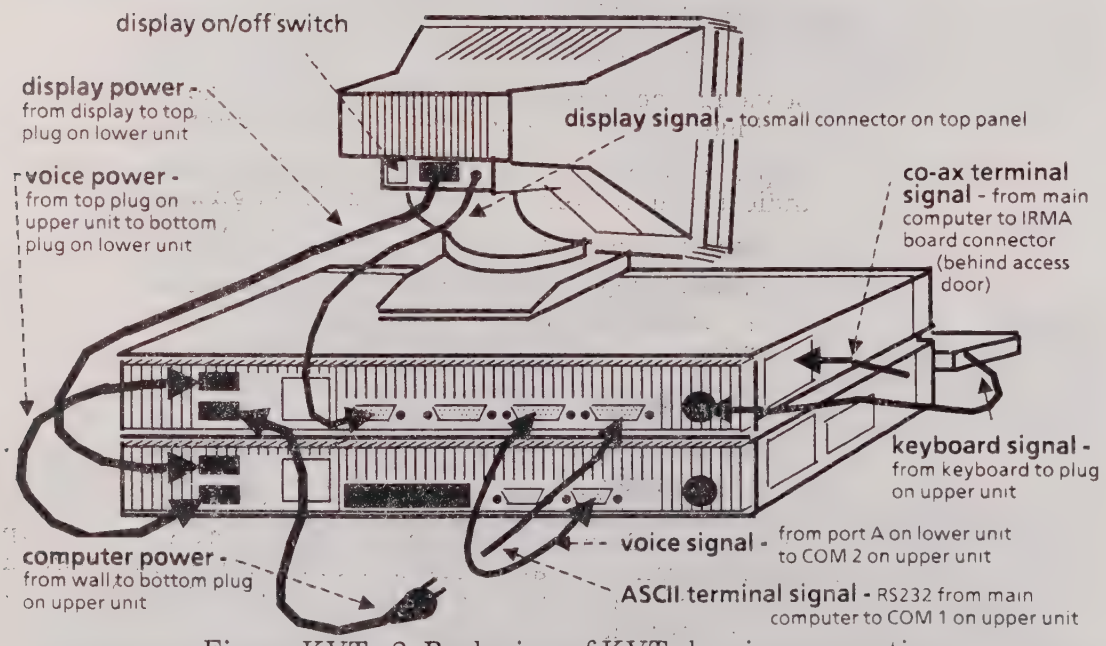
display on/off switch
display power - from display to top plug on lower unit
voice power - from top plug on upper unit to bottom plug on lower unit
display signal - to small connector on top panel
co-ax terminal signal - from main computer to IRMA board connector (behind access door)
keyboard signal - from keyboard to plug on upper unit
voice signal - from port A on lower unit to COM 2 on upper unit
computer power - from wall to bottom plug on upper unit
ASCII terminal signal - RS232 from main computer to COM 1 on upper unit

Figure KVT - 2: Back view of KVT showing connections

4. Next hook up the four signal cables (see Figure KVT-2):

- Connect the 9-pin end of the **voice signal** cable (white RS232 cable) to **PORT A** on the voice unit. Connect the 25-pin end to **COM 2** on the computer.
- The **display signal** cable is permanently attached to the display; plug the other end into the computer connector labelled **MONO DISPLAY.**
- The **keyboard cable** is permanently attached to the keyboard. Plug the other end into the round socket on the far right of the computer.
- Plug the microphone cable into the front of the voice unit. After raising the keyboard to its normal (tilted) position, slide the cable under the back of the keyboard to put it out of the way.

Power On

For proper operation, the computer's power ON/OFF switch must control power to all three units of the KVT. Making correct cable connections will achieve this. Also, after switching on power to the computer, you must turn on power to the DISPLAY and leave it on. This is only done once, at installation. Power to the display will be controlled by the computer power switch after that. Follow this procedure:

- Locate the power ON/OFF switch at the right front of the computer and switch power ON.
- Locate the DISPLAY power ON/OFF switch at the rear of the display ; switch power on and leave it on.

These events happen when power is applied:

- DOS boots up, the voice unit software is downloaded, the KVT pop-up window appears, and then the DOS prompt (**C:>**).
- Words spoken into the microphone cause the green speaking-level indicator on the front of the voice unit to light.
- Keys pressed on the keyboard are shown on the display.

If your KVT is intended for use as a 3270 or ASCII terminal, go on to the next steps. Otherwise skip to "Using the KVT as a Voice Controlled Terminal."

Starting 3270 Terminal Operation

Referring to Figure KVT-2, locate the IRMA board connector on the right side of the computer behind the access door. Connect the *co-ax* terminal signal cable from your main-frame computer to the IRMA board connector. Now refer to the IRMA manual to set up communications parameters and establish communication between the KVT and mainframe computer. Skip to "Using the KVT as a Voice-Controlled Terminal."

Starting ASCII Terminal Operation

Referring to Figure KVT-2, locate the 25-pin connector labelled COM1 at the rear of the computer. Connect the RS232

communications line from your main-frame computer to **COM1**. Now refer to the CROSSTALK manual to set up communications parameters and establish communication between the KVT and main-frame computer.

USING THE KVT AS A VOICE-CONTROLLED TERMINAL

Once you have established communications, you are ready to train the KVT to your voice, make vocabularies and use them to control applications programs. There are three further steps in using the KVT:

- Enroll as a user (see Chapters 3-5 in this manual)
- Make a vocabulary. (see Chapters 6-9 in this manual) An applications specialist may have already made a vocabulary for your application. If so, go directly to the next step.
- Train the vocabulary to your voice (see Chapters 10-13 in this manual).

RESTORING THE PC HARD DISK

In the event that information on the hard disk is lost, restore the pre-installed software by following this procedure:

1. Put MS-DOS on the hard disk. To do this, read "Preparing the Hard Disk" in Chapter 1 of the *WYSEpc User's Guide*. This section tells you how to initialize, create and format an MS-DOS partition on the disk
2. Put KVS System Software on the hard disk. To do this, read "Installing System Software" in Chapter 2 of the *KVS User's Manual*. This section tells you how to install the 3-floppy set and the demonstration disk, WORDKIT.
3. You can now restore your phrase books and language models from backup floppies using the KVSMOVE RESTORE program. To do this, read "Phrase Book Backup:KVSMOVE" in Chapter 4 of the *KVS User's Manual*. If you have a stand-alone KVT, disk restoration is now complete. If you have an ASCII terminal, skip to step 5.
4. If you have a 3270-compatible terminal, the next step is to install the IRMA software. Here's how:
   1. At the C> prompt, type MKDIR \IRMA [RET].
   2. Place each of the three Irma disks and the KVS Irma disk in the floppy drive in turn; at the C> prompt type,

      COPY A:*.* C: \IRMA[RET]

      to copy each onto the hard disk.
   3. Add the IRMA directory to the datapath by typing the following AUTOEXEC.BAT file at the C> prompt:

      COPY CON AUTOEXEC.BAT [RET]

      PATH C:\;\KVPROG;\DOS;\IRMA;\ [RET]

DMA64 [RET]
PROMPT PG [RET]
KVLOAD/N [RET]
[F6] [RET]

Proceed to step 6.

5. If you have an ASCII-compatible terminal, the next step is to install CROSSTALK software. Here's how;
   1. At the C > prompt, type MKDIR \XTALK [RET].
   2. Place the CROSSTALK disk in the floppy drive and type

      COPY A:*.* C:\XTALK[RET]

   to copy the floppy onto the hard disk.

   3. Add the XTALK directory to the datapath by typing the following AUTOEXEC.BAT at the C > prompt:

      COPY CON AUTOEXEC.BAT [RET]
      PATH C:\;\KVPROG;\DOS;\XTALK;\ [RET]
      DMA64 [RET]
      PROMPT PG [RET]
      KVLOAD/N [RET]
      [F6] [RET]

6. Locate the power ON/OFF switch at the right front of the computer; switch power OFF and then switch it ON again. See "Power ON," page D-5 for the events that happen when power is applied.

? 12-10, B-4, B-5, B-6, B-8, B-9
< > 5-7, 8-6, B-7
\123files directory 2-10ff
\\a 8-12, 8-17
! commands 4-11, 8-4, B-4, B-8, B-9

A
<acoustic> B-4
acoustic modeling 13-5
add phrases 12-2, 12-14
    voice command 8-28, 8-30ff
    when to 11-3
    with #EDIT 8-24
add tokens 10-6, 10-7ff
    voice command 8-28, 8-32ff
    when to 13-4, 13-5
    with #VOICE_COMMAND 8-26ff
angle bracket commands B-3
    command summary B-4ff
    in translation strings 8-4
    and #VOICE_COMMAND 8-26
application language model
    creating 8-2ff
application program
    flow of 8-8
    integrating the KVS with 1-3
    KVS transparent to 1-2
    training 4-2
    vocabulary for 4-2, 8-2
application requirements
    for language model design 8-10
ASCII format language model 6-2,6-5ff, 9-4
    converting to KVS format 9-5ff
AUTOEXEC.BAT 2-7, 14-2ff, 14-3ff
AUTOLOAD.BAT 14-3ff

B
background noise 4-6, 4-14
    during enrollment 13-5
    minimizing effects of 13-8
    and recognition 13-9
    and training 13-8
back quote B-7
    with nonprinting characters 8-5
    for use with <edit> 8-6, 12-11
backslash
    for nonprinting characters 8-4, 8-5
    when to double 8-5
"back three" 8-28, 8-39
backup
    of language model 4-15, 14-7
    of phrase book 4-15ff, 9-3, 9-4
<banner> B-4
batch file
    creation of 14-5
BDIST parameter 13-12, B-6
<beep> B-4
boilerplate paragraph 8-29, 8-44
branching 8-29, 8-47
BSCORE parameter 8-40, 13-12, 13-13, B-6
<bye> 5-7, 5-8, 9-11, 10-5, B-4

C
capacity (*see* vocabulary capacity)
capabilities of KVS 1-2
<!choose> 8-4, 8-5, 8-26, 8-30, 8-32, 8-34, 14-4, B-4
choosing spoken phrases 8-2
commands for host utilities, App. B
commands for KVS control, App. B
commands for EDLIN B-11
COM1 and COM2 (*see* communications port)
communications
    between host and KVS (*see* KVRESYNC)
communications port 2-4 and 2-6 (*figs.*)2-13, 2-14, 14-3, C-1,C-2ff
computation
    to create voice profile 4-3, 4-12
    of phonetic model 13-6
computation status 4-13
condense LM switch 12-32
CONFIG.SYS 2-7, 2-14
confirm mode 13-13ff, B-5
conflicts
    communications C-2ff
    between host and KVS software C-2
    keyboard C-4
confusable phrases 8-8, 8-11, 10-7, 11-3, 11-4ff, 13-2
    eliminating before phonetic modeling 13-6
correcting typing mistakes
    in <edit> 7-13, 12-10
correcting recognition errors
    with voice commands 5-6, 8-18, 8-20, 8-21, 8-22, 13-11
count function 10-3, 10-8, 12-31
CTRL-C (clear) 7 -13, 12-10
CTRL-G (get) 7-13, 12-11
CTRL-P (put) 7-13, 12-10
CTRL-W (wildcard) 7-14, 12-10, 12-12, 12-
cursor movement keys
    in <edit> 12-9ff
customizing user environment 14-2ff

D
data collection 4-8, 12-5
    setting up phrase book for 12-30ff
<datapath> B-4
DATES language model 7-2, 7-6, 7-8
delay of message transmission (*see* <!wait>)
delete characters
    in <edit> 12-10
"delete last word" 5-6
#DELETE LAST WORD 8-18, 8-20, 13-11
delete tokens 10-8, 13-6
DEVICE command (*see* DOS DEVICE)
disk capacity 1-3, 2-7, 2-11
DOS COPY command 4-16
DOS DEVICE command 2-14
DOS DATE command 7-2ff
DOS DIR command 7-2ff
DOS PATH command 2-8, 2-13
DOS SET command 14-3, B-2, B-3
DOS bug
    \KVPROG to fix C-4
<drive> B-4
drive 14-4
dual floppy system
    and enrollment 2-7

E
#edit subset 8-8ff, 8-19
#EDIT example 8-24ff
<edit> command 2-9, Chap. 12, B-5
    features of 12-5ff
    command summary B-9ff
    compared with spreadsheet program 6-5, 6-7
    files menu 12-8
    functions 12-13ff, 12-8(*table*)
    language model compared with ASCII format 9-2, 9-4(*table*)
    leaving 12-12
    main menu 12-8
    for making language model 7-6ff
    for making phrase book 6-3, 12-6
    for changing phrase book Fig. 12.1
    options 12-9ff (*table*)
    setup menu 12-8
    vocabulary list menu 12-8
    voice recognition with 8-9ff, 8-19, B-6
editing options 12-9ff (*table*)
    listed by <edit> 12-5
EDLIN commands B-11
ENROLL.KAM 4-12, 4-14, 9-3 and 9-4 (*tables*), 13-5ff (*see also* voice profile)
ENROLL.KPL 4-14, 9-3 and 9-4(*tables*)
ENROLL.KPM 4-12, 4-14, 9-3 and 9-4 (*tables*), 13-5ff (*see also* voice profile)
ENROLL vocabulary (*see* vocabulary)
enrollment 1-5, 1-6, Chap. 4
    computation 4-3, 4-12
    on dual floppy system 2-7
    finishing 4-11
    interrupting 4-9
    recording tokens 4-7
    resuming 4-10
    status of computation 4-11
    testing Chap. 5
enrollment data
    improving 13-5
    questionable 4-9
*ENTER 7-11, 8-12, 8-13
*EXIT 8-12, 8-14
error correction (*see* correcting recognition errors)
error messages
    during backup 4-15ff
    ENROLL 4-12
    INSTALL 2-11
    KVLOAD 2-12ff
    during training 10-9
error prevention 13-2
export.prn file format 6-6, 7-21(*fig.*), (*see also* input.prn file)

F
[f]ile [i]mport 7-15
FORMAT failure C-4
FORTE C-4

G
gain 4-6, 13-9
    adjusting 4-6
    and recognition 5-3
    setting 3-7

H
hardware and software compatibility App. C
hardware installation 2-3ff
headset
    adjustment 3-2, Fig.3-2
    care 3-5
help messages
    when to supply for user 8-50(*table*), 8-51
<help> B-5
homonyms 11-3, 11-5
    for translations 8-29
host resources 1-3
host utilities Chap. 9
    commands for App. B

I
*ILLEGAL 8-12, 8-16, 11-8
INSTALL 2-7ff, 14-2, B-1, C-4
  error messages 2-11
IRMA C-4, App. D
import.prn file format 6-5(*fig.*), 6-6, 9-9
  9-4(*fig.*)
importing words from WORDKIT 7-15
including subsets from WORDKIT 7-6, 7-8ff
incompatible programs C-2
input.prn file 9-5
installation of KVS 1-6, Chap. 2
  cable connections 2-4, 2-6(*fig.*)
  contents of packing carton 2-2
  with fixed disk PC 2-7
  hardware 2-3ff
  INSTALL program 2-7ff
  noise levels 2-3
  physical requirements 2-2
  power requirements 2-2
  software 2-7ff
  temperature and humidity 2-3
  WORDKIT 2-9ff, Chap. 7
invalid characters in phrase book name
  12-22

K
.kam 4-12, 4-14, 9-3, 9-4
.klm 4-14ff, 9-3, 9-4, 9-5
.kmp 4-14ff, 9-3, 9-4, 9-5
.kpl 4-12, 4-14, 9-3, 9-4
.kpm 4-12, 4-14ff, 9-3, 9-4, 13-5ff
.kvc 4-14ff, 9-3, 9-4
KV3DR.SYS 2-14, 14-2
KVCNTRL 14-3, B-1
KVENROLL 4-2, 4-5, 4-12, 9-3, 9-4, 13-8,
  B-1
KVLISTLM 6-5, 6-6, 9-2, 9-6ff, 12-3, B-2
  for changing from <edit> to ASCII
  format 9-8ff, Fig. 12.2
  sample output 9-7
KVLOAD 2-11ff, 2-15, B-2
  error messages 2-12ff
KVMAKELM 6-5, 6-6, 7-20, 9-5ff, 12-3,
  Fig. 12.2, 13-4, 14-6ff
KVMAKEPB 6-6ff, 9-2, 9-9ff
KVMODPB 9-2, 9-10ff, 12-3, 12-8(*fig.*), 13-4
KVPORT 2-13, 2-14, 14-3, C-1
\KVPROG directory 2-8, 2-14, 14-2, C-4
KVRESYNC 2-15, B-2
KVS application program (*see* KVTEST)
KVSEND 14-4, 14-5, B-2
KVS hardware 1-4, 1-6
KVS initialization
  automatic 14-2
KVS format language model 6-2, 6-4, 9-3
\KVSLMOD directory 2-9, 2-10, 7-2, 9-4, 9-5
  12-24, 14-7
KVS-mainframe communication C-1
KVSMOVE 4-15ff, 14-7, B-2
KVSPEED 14-3
KVSPEEDA 2-14
KVS performance 11-2ff, 13-2ff
KVS start-up 2-11ff
  automatic 14-2
\KVSUSERS directory 4-13, 4-15, 9-2, 9-3,
  14-3ff, 14-6
KVS utilities 2-7, Chap. 9, App. B
KVS window 2-12(*fig.*), 5-7, B-7 (*see also*
  LISTEN parameter)
  blanking 8-13
  keeping on screen 8-13
  recognition in 8-8ff
  for user messages 8-12, 8-50(*table*)
KVT App. D
KVTEST 5-2ff, 10-2, 10-6, 11-5ff
KVTRAIN 5-9, 10-2, B-2

L
language model 1-2, 1-5, 2-9, 6-2, Chap. 7,
  7-2ff, 9-2,
  application requirements 8-10
  changing 12-2, 12-3ff
  changing with <edit> 12-2ff, 12-5
  changing with spreadsheet program
  12-3
  design considerations 8-10ff
  design of 6-2, Chap. 8
  design for ease of use 8-10ff
  elements of 8-2ff
  interrupting creation of 6-4
  KVS requirements 8-11
  listing with <edit> 12-26ff
  listing with KVLISTLM 9-6ff
  making with <edit> 7-6ff
  making with text editor 7-20
  making with spreadsheet program
  7-16ff
  printing with KVLISTLM 9-6ff
  publishing 12-24
  reporting size and content 12-5, 12-26f
  searching with wild cards 7-14
  size of 8-9
  storing on fixed disk 14-5
  transferring via fixed disk 14-6
  transferring via floppy disk 14-7
language models
  on WORDKIT disk 2-9ff, 7-2

learn mode
in <edit> 12-11
leaving <edit> 12-12
LISTEN parameter 8-9ff, B-6 (*see also* KVS window, #edit, #voice_command)
#LISTEN TO ME 7-11, 8-18, 8-23, 8-38
and <pause> 8-23
"listen to me" 5-7
Lotus 1-2-3 (*see* spreadsheet program)

M
main menu
in <edit> B-9
MAXPHRASES parameter 13-12, B-7
MAXRATIO parameter 13-12, B6
memory capacity 10-5
how to determine 8-9
menus
<edit> functions 12-8
files 12-8, B-9
main 12-8, B-9
set-up 12-8, B-10
vocabulary list 12-8, B-10
MIC-LINE switch 2-15(*fig.*)
microphone
turning on and off 5-7, 12-11
turning on with #LISTEN TO ME 8-23
<microphone> B-5
microphone check 3-2, 8-37 (*see also* gain)
bypassing 5-8
to model background noise 13-8
during enrollment 4-6ff, 4-12, 4-14
and recognition 5-3
repeating 3-2, 4-8, 5-8
during training 10-3, 1-4, 10-5
using <pause>
"microphone check" 8-28, 8-37
microphone jack 2-4(*fig.*)
"microphone off" 8-28, 8-38
microphone position 3-3ff, 3-4(*fig.*), 4-8,
to eliminate background noise 13-8
multiple commands (*see* voice macro)

N
NEWDATES language model 7-2ff
adding phrases and translations with <edit> 7-11ff
adding phrases and translations with spreadsheet program 7-16ff
creating with <edit> 7-6ff
editing translations 7-11
excerpt after copying subsets 7-10
excerpt before copying subsets 7-9
listing of 7-3ff
saving 7-14ff, 7-20
shortcuts to adding phrases and translations 7-13
training 7-6ff, 7-15
"next choice" 5-6
#NEXT CHOICE 8-18, 8-21, 13-11
nonprinting characters App. A, 12-11
C abbreviations for 8-5, App. A
providing window messages for 8-41
in translation strings 8-4ff

O
octal notation
for alphanumeric keys 8-6, App. A
for nonprinting characters 8-4, 8-5
for translations with spreadsheet program 8-6
ordered_list output mode 11-4ff, 3-12, B5
<output> 13-13, B-5
output modes 11-4, 13-12, 13-13, B-5

P
"paragraph" 8-28, 8-43
<parameter> 13-11, B-6
pause between words 3-5ff
shortening (*see* wordgap)
"pause" 8-28
<pause> 5-7 , B-7
for microphone check 5-7
restarting after 5-7
pausing during recording tokens 4-9, 10-5
"period" 8-29, 8-43
<phonetic> 13-6ff, B-7
phonetic model
for application vocabulary 13-6
phrase
completion in <edit> 12-10 (*see also* ?
defined 3-6
duration 8-3, 8-9, 8-11
phrase book 1-5, 1-6
application 5-9, 9-2, Chap. 10
changing Chap. 12, 12-3 and 12-4(*figs.*
changing after training 9-10ff, 12-25
defined 1-5, 6-2
deletion 12-23
directory listing 12-21, 9-6ff
listing contents 12-26ff
merging 12-25
new 12-6ff
saving 12-22
size 12-26
storing before training 9-9
trained 9-3, 9-4
training for application Chap. 10

untrained 9-3, 9-9
phrase book creation Chap. 6, 9-9, 12-6ff
from ASCII language model 9-4(*table*)
with <edit> 6-3, 6-4, 9-2ff, 9-3(*table*)
interrupting 6-4
KVS format and ASCII format compared 6-4
from published language model 12-6
with spreadsheet program 6-3, 6-5, 9-5ff
spreadsheet program and <edit> compared 6-5
with text editor 6-3, 6-7
phrase book file management 12-21
phrase book files 4-14(*table*)
phrase book name
invalid characters 12-22
"phrase books" 8-50
phrases
adding 12-14, 12-25
deleting 12-15
distinctive 13-2
duplicate 12-25
editing 12-5, 12-16
listing 9-8, 12-27, 12-29
short 4-12, 11-4
similar-sounding 11-5
synonyms 8-3
valid 8-3
phrase-translation pairs 7-2, 7-11, 7-17ff, 8-4
listed by KVLISTLM 9-6
power off 2-4(*fig.*), 2-15
power on 2-4(*fig.*), 2-11ff
*PREFIX 8-12, 8-17
.prn files 7-21, Fig 7.1, 9-5, 9-9(*fig.*)
on WORDKIT disk 2-9
problem phrases 11-4, 11-7
<profile> B-7

Q

"quit" 8-28, 8-41
quote characters (*see* back quote, backslash)

R

recognition
and background noise 13-9
microphone check 5-3
setting gain 5-3
stopping and resuming 5-6ff, 5-8
testing 5-2ff, Chap. 11
recognition accuracy 4-8, 5-8, 9-5ff, 11-2, 11-4, 12-3
improving 10-6 and 12-2(*tables*)
improving in language model design 8-11
improving with enrollment 13-5ff
improving with phonetic modeling 13-6ff
improving with rejection threshold 11-8, 8-28
improving with subsets 8-8
improving with token editing 13-4
improving with token resolution 13-4
recognition candidate list 11-7
parameters for 13-11ff, 13-12(*table*)
recognition errors
correcting 13-11
recognition score 10-6, 11-5
improving 10-6(*table*)
rejection threshold 11-8
<recognize> 5-3, 5-7, 5-8, 10-6, 11-2 12-2, B-7
recording tokens (*see also* tokens, recording) 4-8ff, 10-3ff
interrupting 4-10
pausing 4-10
pausing and resuming 4-10, 4-11, 10-5
reference tokens (*see* tokens)
rejection threshold (*see* <threshold>)
adjusting 11-8
remember key (*see* CTRL-P) 12-9, 12-10
repeat keys (*see* TAB, CTRL-G) 12-9, 12-10
"replace tokens" 8-28, 8-34ff
replace tokens 10-6, 10-7
when to 11-3, 11-4, 11-6, 13-6
resolution (*see also* token resolution) 8-10, 9-5ff, 12-30, 13-4
<restart> 2-15, 12-3, B-8
restarting the KVS 2-15
Run-Time Library 1-3, 13-13

S

saving
enrollment 4-10
language model 7-14ff, 7-20
phrase book to disk 10-3
phrase book editing 12-5
training 10-5
search function
in <edit> 12-10
serial communications port (*see* communications port)
SET 14-3, B-2, B-3
"set threshold" 8-28, 8-40
set-up of KVS 2-2, 2-3
set-up for recognition 1-5
sharp (#) phrases 8-6ff, 8-18(*table*) 13-11
sharp subsets 8-8ff, 8-18(*table*), 8-24, 8-26

shortcuts
for language model creation 12-5
short phrases 4-12, 11-4
signal lights 2-4(*fig.*) (*see* speaking-level lights)
speaking-level lights 2-15, 3-7, 4-7, 5-8, 13-9
speaking rate 3-6
speaking tips 3-5ff
speaking volume 3-7
special characters
in <edit> B-10
speeding up digit entry 13-9
spoken phrases
adding application-specific 8-2ff
choosing for an application 8-2ff
spreadsheet program 2-9
compared with <edit> 6-5, 6-7
for phrase book creation 6-5, 7-16ff
star (*) phrases 8-6ff, 8-12(*table*)
start-up of KVS 2-11ff
<status> 4-11, B-8
<subset> 5-4, 7-5, B-8
for switching subsets 7-6
subsets 5-4ff, 8-7
active 8-8
to avoid confusable phrases 13-2ff
changing names 12-19
checking for translations 11-7
changing by voice command 8-8
Command in NEWDATES 7-2, 7-3
copying 12-20
copying in new language model 7-9ff
defined 8-7
define_living 5-4ff
Digit in NEWDATES 7-2, 7-3
editing with <edit> 12-5, 12-13ff
for improving recognition accuracy 8-8
listed by KVLISTLM 9-6
listing 12-28, 12-29
living 5-4ff
lower 5-4ff
making new 12-17
Months in NEWDATES 7-2, 7-4
for multiple translations 8-3, 8-7, 8-11
removing 12-18
rules for creating 8-7
switching 7-6
upper 5-4ff
Yrslong in NEWDATES 7-2, 7-4
subset living
ENROLL vocabulary for 5-5
*SUFFIX 8-12, 8-17
synchronizing communications (*see* KVRESYNC)
synonyms 8-3, 10-7
for voice commands 8-29, 8-46

T
TAB 7-13, 12-10
testing recognition 4-3, 5-2ff, 6-4, 8-28, 8-36 Chap. 11
text editor 6-7, 7-20ff
threshold (*see also* rejection threshold)
lowering and raising 8-28, 8-40
<threshold> B-8
token memory 10-4, 10-5
token resolution 8-10, 9-5ff, 12-30
changing 13-4
tokens 1-2, 4-2, 9-2
adding 8-32
adding, deleting, replacing during training 10-6ff
bad 4-7, 4-9, 4-12ff, 10-4, 10-6(*table*), 11-5ff 13-9
changing 12-2
counting 10-8
deleting 10-8
editing with #voice_command
good 4-3, 4-6, 4-8, 11-5ff
interrupting recording of 4-10
listed in KVLISTLM 9-7
making 4-3
number of 9-7, 10-7
number of during training 12-31
pausing recording of 4-10
recording 3-7, 4-3, 4-6, 4-8ff, 10-3ff
replacing 4-9, 8-34, 10-8
saving 4-10, 10-5
skipping during recording 10-4
testing recognition of 10-6, Chap. 11
training 10-2ff
varying order during training 6-6, 9-6
top_choice mode 11-6, 13-12, 13-13
<train> 7-15, B-8
training 1-5
application vocabulary 5-9, Chap. 10
and background noise 13-8
compared with ENROLL 10-4
new language model 7-6ff, 7-15
starting with ASCII language model 10-2
starting with <edit> 10-2
status of 10-5
"train mode" 8-28, 8-42
"train tokens" 8-26ff, 8-28
translations 1-5, 1-6, 5-4
defining 8-4
editing 7-11, 12-5, 12-13ff

listed with KVLISTLM 9-6ff
listing with <edit> 12-29
multiple in language model 7-5, 8-3
sequence of sending 8-4
valid 8-4
transparent mode 1-2
"try again" 5-6
#TRY AGAIN 8-18, 8-22, 13-11
tutorial
making language model with <edit> 7-6ff
making language model with spreadsheet program 7-16ff
making language model with text editor 7-20ff

U
UNINSTALL B-3
*UNKNOWN 8-12, 8-15, 11-8
unpacking the KVS 2-2
<user> 5-3, B-9
user confirmation 8-48
user help commands 8-50(*table*)
user start-up
automatic 14-3, 14-4
utterance duration (*see* phrase, duration)

V
valid phrases 8-3
valid subset names 8-7
valid translations 8-4
<version> B-9
vocabulary 1-5
adding and deleting 12-5
application versus ENROLL vocabulary 13-7,
changing with <edit> 13-4
changing with spreadsheet program 13-4
constraints on 8-3
ENROLL 4-2, 4-3, 4-4(*table*), 4-6, 5-2
testing 11-5
vocabulary capacity 1-3, 8-9ff (*see also* resolution)
voice application mode 13-13ff, B-5
VOICECMD language model
on WORDKIT disk 2-10
#voice_command subset 8-8, 8-9
#VOICE_COMMAND example 8-19, 8-26
voice commands
for correcting recognition errors 13-11
problems with execution 11-4
for switching subsets 7-6, 11-8
and translations 8-28(*table*)
for user help 8-50(*table*)
voice correction 5-6, 13-11
voice level 4-6
voice macro 8-29, 8-45, 13-10
voice profile 1-5, 4-2ff, 4-12, 4-14ff, 6-4, 9-2, 9-3 and 9-4(*tables*)
accuracy 4-2
computation of 4-3
improving 13-5
other than ENROLL 9-10, 13-7ff, 12-5, 12-33

W
<!wait> 8-29, 8-49, 11-9, B-9
<window> B-9
.wks file
on WORDKIT disk 2-9
WORDKIT 2-2, 2-9ff
contents of disk 2-9ff
for creating new language model 7-2, 7-6, 7-8
sample output 9-7
using with spreadsheet program 7-15ff
"where am I" 8-51
wild card [CTRL-W] 7-8, 7-14, 12-5, 12-10, 12-12
for searching language model 12-12(*table*)
word (*see* phrase)
WORDGAP parameter 3-6, 13-9

KVS PROBLEM REPORT

KVS SERIAL # _____ SOFTWARE VERSION 1.0 - OCT. 1985	YOUR NAME _____ COMPANY _____
DATE:	ADDRESS _____
HOST COMPUTER:	_____ PHONE

GENERAL NATURE OF PROBLEM: [ ] Hardware [ ] Software [ ] Documentation
[ ] OTHER:

SPECIFIC PROBLEM: (it is very helpful to know what happened just before the problem)

RECOMMENDATIONS:

RECEIVED: _____ KVS SEQUENCE # _____ ASSIGNED TO: _____

KURZWEIL APPLIED INTELLIGENCE, INC. - Attn. Frank Tadley
411 Waverley Oaks Rd.
Waltham, MA 02154 - 8465
USA